UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

Sistema de apoio ao ensino de diagramas de actividades da UML – Decisões e cenários alternativos

DISSERTAÇÃO DE MESTRADO EM
INFORMÁTICA

CARLOS RODRIGUES PEREIRA MATIAS

Vila Real, 2009

Sistema de apoio ao ensino de diagramas de actividades da UML – Decisões e cenários alternativos

Dissertação de Mestrado apresentada por

Carlos Rodrigues Pereira Matias

Sob orientação do Professor Doutor João Eduardo Quintela Alves de Sousa Varajão

e co-orientação do Professor Doutor João Manuel Pereira Barroso

Universidade do Trás-os-Montes e Alto Douro

Departamento de Engenharias

Vila Real, 2009

Aos meus pais, Adília e Dinis.

Agradecimentos

Agradeço aos meus pais, pela educação que me deram, pelos valores que me incutiram, pelo esforço que o meu percurso lhes exigiu e pela dedicação e insistência no meu sucesso e felicidade. Aos meus irmãos pela ajuda e apoio que, cada um à sua maneira soube e se prontificou a dar. Aos meus sobrinhos pela cor que dão à minha vida.

Ao Professor Doutor João Eduardo Varajão por ter sido um orientador atento, preocupado e incentivador, pelo acompanhamento delineado que fez do meu percurso e por tê-lo tornado um percurso de constantes aprendizagens e vitórias, finalmente pela compreensão e amizade que demonstrou dentro e fora da sala de aula.

Ao Professor Doutor João Barroso pelo reconhecimento das minhas potencialidades e por ter contribuído para que as mesmas se traduzissem em conhecimento e experiência, pelo acompanhamento e preocupação na obtenção de bons resultados e pela amizade demonstrada no apoio dado a vários níveis ao longo dos último cinco anos.

À Universidade de Trás-os-Montes e Alto Douro (UTAD) e a todos os professores que acompanharam de perto o meu percurso e que contribuíram para a bem sucedida chegada a mais esta meta.

Ao Professor Doutor Ricardo Martinho pela disponibilidade, opinião e conselhos que no decorrer deste estudo demonstraram ser importantíssimos. Aos alunos dos cursos de Licenciatura em Informática e de Licenciatura em Tecnologias da Informação e Comunicação da UTAD, pela participação nas experiências realizadas ao longo desta investigação.

A todos os meus amigos, em especial ao Nuno Cruz por ter sido um verdadeiro colega, atento e trabalhador, pelo apoio em todos os momentos desgastantes que nos perseguiram e pela amizade e carácter que lhe reconheço. Agradeço também com especial atenção ao Marco Canelas, à Cidália Pereira, à Susi Assunção, ao Armindo Fernandes e ao Luís Portela pela prontidão, interesse e preocupação demonstrados e pela amizade que tenciono manter sempre. A todos os restantes, sei que posso contar convosco e retribuo a amizade.

Finalmente, quero agradecer com especial carinho, à Sónia, pelo constante incentivo, pela atenção, preocupação e compreensão que fazem dela uma mulher especial e pela força que mesmo não tendo para si, sabe transmitir a quem precisa.

A todos, *Muito Obrigado!*

Resumo

Na actualidade, com o constante desenvolvimento das Tecnologias da Informação, verifica-se uma forte presença das mesmas nas organizações e até nas nossas casas. Por questões de sobrevivência e competitividade do mercado, as organizações já dependem em larga escala dos seus Sistemas de Informação e para que se mantenha uma resposta adequada ao mercado é fulcral melhorar constantemente a qualidade do software. Para isso é extremamente importante fazer uso de técnicas de planeamento e modelação de software e assim evitar falhas e custos de desenvolvimento e melhorar o cumprimento dos prazos de entrega dos mesmos.

Dos vários tipos de diagramas disponibilizados pela Unified Modeling Language (UML), os diagramas de actividades são dos mais recentes, mas já reconhecidos pela UML como um tipo de diagramas útil e fiável na modelação de software, o que traduz a importância da inclusão dos diagramas de actividades no ensino da Engenharia de Software.

Tal como no ensino de vários dos diagramas da UML, também no ensino dos diagramas de actividades se verificam dificuldades por parte dos alunos, ao nível da compreensão dos mesmos e dos seus objectivos, muito por culpa dos currículos dos cursos onde a UML é estudada, que por várias razões não disponibilizam o tempo necessário para a prática do desenvolvimento de diagramas de actividades.

Por outro lado, a simulação de casos reais em ambientes virtuais, recorrendo ao mais recente software, tem demonstrado melhorias na aprendizagem e aperfeiçoamento de técnicas, em menos tempo e com mais motivação dos alunos que estudam determinada área. Por este facto, fica clara a importância de desenvolver ferramentas de simulação que permitam apoiar o ensino dos diagramas de actividades, familiarizando os alunos com a técnica e incutindo-lhes a importância da sua utilização na modelação de software.

Assim, surge a oportunidade de desenvolver um sistema de apoio ao ensino de diagramas de actividades da UML, que contemple as regras e sintaxe da mesma. Neste caso específico deu-se especial interesse ao estudo das decisões e cenários alternativos presentes nos diagramas de actividades, tentando fazer um correcto levantamento da sua utilidade e forma de facilitar o seu entendimento por parte dos alunos da Engenharia de Software.

Abstract

Nowadays, with the constant development of Information Technology, there is a strong presence of it in organizations and even in our homes. For reasons of survival and competitiveness of the market, organizations now depend on a large scale of its Information Systems, to keep an adequate response to the market is key to constantly improve the quality of software. Thus, it is extremely important to use planning techniques and software modeling to avoid failures, increased development costs and to improve delivery deadline compliance.

Of the various types of diagrams available for the Unified Modeling Language (UML) activity diagrams are the most recent, but already recognized by the UML as a useful and reliable type of diagrams in software modeling, which shows the importance of including activity diagrams in Software Engineering learning.

As in several UML diagrams, difficulties may occur in understanding activity diagrams by the students, in comprehending the diagrams and their objectives, much to blame on the curricula of courses where the UML is studied, which for various reasons do not provide the necessary time to practice the development of activity diagrams.

Furthermore, simulating real cases in virtual environments, using the latest software, has shown improvements in learning and refinement of techniques in less time and with greater motivation of students who study the area. For this, it is important to develop simulation tools to support teaching of activity diagrams, familiarizing students with the technique and instilling them the importance of its use in software modeling.

Thus arises the opportunity to develop a system to support teaching UML's activity diagrams, which includes its rules and syntax. In this particular case we gave emphasis to studying decision and alternative scenarios in activity diagrams, trying to make a proper survey of its usefulness and how it facilitates understanding by students of Software Engineering.

“Sorte é o que acontece quando a preparação encontra a oportunidade.”

Elmer Letterman

Índice Geral

Índice de Tabelas

Índice de Figuras

Siglas

Nesta dissertação são utilizadas abreviaturas de designações comuns, apenas apresentadas aquando da sua primeira utilização.

DA Diagrama de Actividades

ES Engenharia de Software

HTML HyperText Markup Language

OMG Object Management Group

OO Orientado ao Objecto

PHP Hypertext PreProcessor

SI Sistema de Informação

TI Tecnologia da Informação

UML Unified Modeling Language

XML eXtensible Markup Language

I

1 Introdução

As Tecnologias da Informação (TI) alteraram as formas de comunicação, com implicações nas relações entre seres humanos, na forma como se adquire conhecimento e deram ainda origem a um novo tipo de economia, modificando estruturas e canais de comércio que se mantinham inalteráveis há centenas de anos (Gouveia, 2006).

As TI modificaram os tradicionais mecanismos do poder, fazendo valer a informação como propulsor da evolução da sociedade, impondo uma nova revolução que está em andamento e cujos efeitos provocados só podem ser comparados aos da revolução industrial (Souto, 2008). Ao contrário da revolução industrial, caracterizada por uma morosa evolução tecnológica onde foram necessários cinquenta anos para se percepcionar o potencial da electricidade e mais cinquenta anos para que esta chegasse à casa da maioria da população, a revolução digital está a acontecer muito rápida e diversificadamente. Um exemplo comparativo que caracteriza a rápida evolução das TI, é o facto da rádio e da televisão, terem demorado 38 e 13 anos respectivamente, para atingirem um público de cinquenta milhões de pessoas, enquanto a Internet alcançou o mesmo número de utilizadores em apenas quatro anos (Crandal, et al., 2001).

A Era actual de globalização tem vindo a dificultar a sobrevivência das organizações, obrigando-as a aumentarem o seu nível de exigência para fidelizarem e manterem a confiança dos seus clientes. Este facto incentivador da competitividade entre as organizações leva-as à

alteração de estratégias e ao aproveitamento de todas as tecnologias emergentes, sempre no sentido de dar a melhor resposta ao mercado actual.

Assim, podemos dizer que as TI são uma ferramenta de trabalho indispensável para as organizações que querem manter-se na vanguarda do desenvolvimento e as provas dadas pelas TI são tais, que a sua utilização alargou-se à sociedade em geral, de tal forma que actualmente estão presentes no quotidiano de qualquer cidadão, independente da idade, cultura ou ramo profissional.

Este facto coloca diariamente o sistema educativo perante novos e inadiáveis desafios, ou seja, devem preparar-se os alunos para viverem numa sociedade informatizada e devem formar-se pessoas que cada vez mais terão de exercer profissões caracterizadas por funções de tipo não repetitivo, onde serão solicitadas, cada vez mais, pessoas com capacidades para resolver problemas, comunicar e auto-actualizar-se em espaços de tempo cada vez mais reduzidos (Educação, 2007).

Neste âmbito, torna-se fundamental fazer uso das TI nas escolas, seja pelo contributo na gestão e comunicação entre as partes envolvidas, seja pelo aproveitamento das suas potencialidades no apoio ao ensino. Para isso, é fulcral desenvolver sistemas que facilitem o entendimento por parte dos alunos, da importância da modelação de software antes da sua construção e no quanto isso pode evitar percalços e desvios no processo de desenvolvimento até à obtenção do objectivo final.

1.1 Enquadramento

Ao longo de várias décadas assistimos à utilização do quadro preto, do giz e do livro como únicos instrumentos de trabalho nas escolas; no entanto, nos últimos cinquenta anos, verificou-se a evolução e aproximação do computador à população em geral, tomando conta de praticamente todas as áreas educacionais.

A utilização dos computadores como meio de comunicação e instrumentos de trabalho está a revolucionar os meios de educação e as tecnologias estão, cada vez mais, a ser utilizadas em acções pedagógicas, o que coloca os professores perante o desafio de rever os paradigmas sobre a educação. A Internet como tecnologia e meio de comunicação, tem dado um forte contributo nesse sentido e, segundo Alava (Alava, 2002) veio possibilitar experiências e actividades pedagógicas inovadoras, o que gera novos conceitos e novos modos de aprendizagem.

Na actualidade, o computador, por se tratar de um instrumento recente e actual, é tido como um importante instrumento de trabalho e educação, nas escolas e em casa. Este facto deve-se à constante evolução e à capacidade de auxiliar o ensino nas mais diversas áreas, bastando para isso fazer uma correcta selecção dos programas. Assim podemos dizer que o poder do computador está fortemente associado ao software.

Neste sentido surgiram também as ferramentas de simulação, aplicações de treino e educação que oferecem interfaces dinâmicas e interactivas onde os alunos podem aplicar conhecimentos, explorar áreas específicas e adquirir experiência na aplicação de técnicas de trabalho, permitindo-lhes assim, desenvolver capacidades de reacção ao ambiente real.

Skrien (Skrien, 2001) definiu a simulação como "uma ferramenta educacional poderosa e utilizada amplamente num sem número de aplicações para o ganho de experiência em determinados processos sem qualquer custo monetário ou efeitos perigosos e assim se poder simular situações do mundo real".

De acordo com Wilson (Wilson, 1978), por simulação entende-se que é "um processo de construção de um modelo de um sistema real e a condução de experiências com esse modelo, com o objectivo de compreender o comportamento de um sistema e/ou avaliar estratégias para a sua operação".

Devido à crescente competitividade, muitas empresas foram forçadas a implementar software para melhorar a eficiência e ao mesmo tempo reduzir custos. A modelação com ajuda de software de simulação tornou-se uma metodologia popular (Doukidis, et al., 1990), com uma ampla gama de aplicações (Gogg, et al., 1993) e usada para encontrar estratégias para o melhoramento do desempenho e para criar processos alternativos aos já utilizados.

Na sequência deste facto, muitos pacotes de software foram desenvolvidos para modelar problemas de simulação (Nikoukaran, et al., 1998). No entanto, torna-se patente a existência de vários requisitos por cumprir, limitações e problemas associados a estes pacotes de software (Hlupic, 2000).

Askari (Askari, et al., 1998) argumentam que as simulações são ferramentas educacionais muito úteis na medida em que disponibilizam ambientes dinâmicos e interactivos nos quais os alunos podem explorar áreas específicas e consequentemente desenvolver uma compreensão qualitativa e quantitativa dos sistemas dinâmicos. Têm sido desenvolvidos cada vez mais pacotes de software de simulação, tanto para resolver problemas específicos como para um uso mais abrangente. Também as linguagens de programação têm vindo a evoluir em grande escala para que, através das capacidades de hardware, se

desenvolva mais e melhor software de simulação que ajude a criar métodos actuais de ensino de determinados temas.

Por outro lado, mesmo sendo o ensino uma área possível de aplicação dos benefícios do software de simulação, ainda não se verifica um aproveitamento desejável de tais ferramentas. Existem excepções que foram identificadas como promissoras, mas os horizontes ainda não se encontram completamente abertos para que o software de simulação se coloque em plano de destaque na educação (Drappa, et al., 2000).

Deste modo, as TI e o software de simulação apontam que, quando utilizados de um modo correcto, podem ser um instrumento precioso no processo de ensino-aprendizagem, tornando-o mais motivante, rápido e eficaz.

Por tudo isto, podemos afirmar que a utilização da simulação como ferramenta educacional poderá ser usada num vasto leque de domínios, de forma a serem alcançados, não só conceitos teóricos, mas também experiência e aperfeiçoamento de técnicas de execução na área em estudo, sem falhas relativamente aos custos monetários e sem efeitos perigosos que poderiam resultar na aplicação dessas técnicas directamente num ambiente real. Desta forma, os alunos podem repetir as experiências, com diferentes abordagens e assim aprofundarem conhecimentos em cada simulação executada.

1.2 Motivações, objectivos e método

O sucesso de um projecto depende de vários factores, sendo um deles a qualidade do planeamento, especificação e documentação antecipados das fases de todo o processo de desenvolvimento, reflectindo-se também essa qualidade na manutenção, evolução e adaptação do sistema desenvolvido, a novas necessidades.

Segundo Booch (Booch, et al., 1996), a habilidade que o ser humano tem para imaginar novas e cada vez mais complexas aplicações será sempre superior à sua habilidade de as desenvolver, e a construção de software com erros é o motivo da maioria das falhas dos projectos de software.

De acordo com Conrow (Conrow, et al., 1997) através da análise de 8380 projectos de software, é indicado que 31% dos projectos foram cancelados e 53% apresentavam graves problemas, com aumentos de 189% e 222% respectivamente, sobre os custos inicialmente previstos e sobre o cronograma inicial, que continham apenas 61% dos requisitos inicialmente propostos.

A maior parte dos erros de um sistema (64%), está relacionada com a fase de análise de requisitos, sendo estes apenas descobertos em etapas mais avançadas tais como a fase de testes. O custo para correcção de um erro encontrado durante a fase de análise equivale a 1/5 do que seria durante a fase de testes e a 1/15 se esse mesmo erro for encontrado quando o sistema estiver em uso (Kotonya, et al., 1996).

Apesar das estatísticas, a expectativa e procura por software mais complexo cresce cada vez mais, pois este é o motor que faz funcionar muitas empresas e organizações, que sofrem consequências graves quando muitas vezes o software utilizado não apresenta a qualidade necessária e foge quase sempre ao inicialmente planeado, aumentando os seus custos. Estes resultados são ainda piores em empresas ou organizações onde não existe um comprometimento mínimo com bons princípios de desenvolvimento (Conrow, et al., 1997).

Por este facto, as empresas estão cada vez mais dependentes dos seus Sistemas de Informação (SI) o que as leva a exigirem que o desenvolvimento dos mesmos seja feito em tempo útil, com qualidade e custos finais iguais aos inicialmente previstos. Desta forma, qualquer criador de software define estes aspectos como sendo os seus principais desafios.

Os principais instrumentos da fase de análise são modelos que representam o problema a ser resolvido num nível de abstracção mais elevado do que a visão computacional, permitindo um melhor entendimento das várias visões do projecto e servindo de base para as etapas seguintes do processo, como a construção do software propriamente dito e os testes.

Revendo a numerosa literatura disponível sobre a Unified Modeling Language (UML), é fácil verificar que, quando comparados com outros diagramas da UML, é dada pouca atenção aos Diagramas de Actividades (DA) na UML 1.x. Isto deve-se basicamente ao facto de na UML1.x os DA serem definidos como um subconjunto dos diagramas de estados (Barros, et al., 2003). No entanto, na UML 2.x, os DA já assumem um papel extremamente importante e são claramente separados dos diagramas de estados (Bhattacharjee, et al., 2009).

Assim, Booch (Booch, et al., 2005) definem os DA como um elemento de modelação simples, mas eficaz, usado para descrever fluxos de trabalho numa organização ou para detalhar operações de uma classe, incluindo operações que possuam processamento paralelo. Por este facto, nem sempre é fácil para os alunos da área compreender os DA, sendo que os aspectos que normalmente provocam mais dúvidas são as actividades concorrentes e as decisões e cenários alternativos.

Foi com base em tudo o que descrevemos anteriormente que se identificou ser útil e interessante desenvolver um sistema que auxilie a compreensão dos DA, a importância dos

mesmos, no âmbito da modelação de um sistema de software e a utilidade destes diagramas aquando da implementação do sistema propriamente dito.

De modo a cumprir as finalidades a que nos propusemos, foram definidos os seguintes objectivos que traduzem as grandes motivações do projecto, respectivamente:

- Identificar o conjunto de características que deverão ser implementadas num sistema de apoio ao ensino de DA da UML, tendo em conta as dificuldades que podem surgir no entendimento do processo[1], por parte de quem as utiliza, nomeadamente na representação das decisões e cenários alternativos de um DA.

- Desenvolver uma aplicação Web para o apoio ao ensino de SI, onde qualquer utilizador poderá criar DA e posteriormente poderá visualizar uma simulação do mesmo.

Este estudo começou pela pesquisa de ferramentas existentes que simulassem os DA da UML, mas que permitissem aos alunos visualizar o processo de negócio descrito em determinado diagrama, de uma forma interactiva e sequencial e fundamentalmente que eliminassem qualquer factor de dúvida durante o acompanhamento da execução do fluxo, nomeadamente a execução de actividades concorrentes e de nós de decisão onde, tal como na realidade, apenas um cenário se sucede aos mesmos. Esta pesquisa não foi bem sucedida, já que não foram encontradas ferramentas que suportassem a visualização adequada da execução de actividades concorrentes e que simplificassem a visualização da execução de cenários alternativos após os nós de decisão. Por este facto decidiu-se desenvolver uma ferramenta que suportasse este processamento e que simplificasse o entendimento geral dos DA por ela executados. Para isso, foram elaboradas experiências laboratoriais, sendo que estas foram estruturadas de uma forma independente, cada uma delas, pensada para avaliar um aspecto/característica e assim conseguir resultados adequados às necessidades dos utilizadores. O planeamento das experiências foi feito passo a passo, sendo a sua maioria planeada de acordo com os resultados da anterior. O momento de análise das experiências pode considerar-se o momento fulcral da presente dissertação, pois foi este que indicou o caminho a seguir aquando da construção da ferramenta de simulação.

Este estudo, de natureza experimental em ambiente controlado, onde o objecto de estudo e as respectivas variáveis estão determinados, baseou-se em experiências realizadas com alunos que nunca tinham tido qualquer contacto com os diferentes diagramas da UML, sendo

[1] Notar que, neste âmbito, o termo *processo* é utilizado para nos referirmos a um processo de negócio ou a um processo de desenvolvimento de software que seja representado num diagrama de actividades.

que todo o trabalho realizado reflecte o conhecimento extraído da observação e análise dos dados retirados dessas mesmas experiências.

1.3 Organização da dissertação

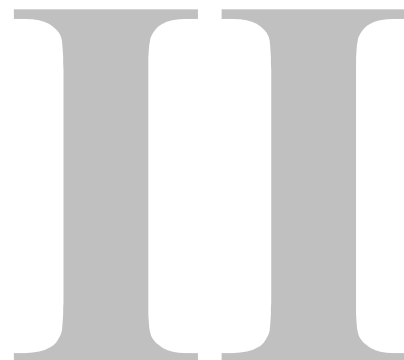
Nesta secção descrevemos sucintamente a estrutura e conteúdo da presente dissertação.

A estrutura foi definida tendo em conta o processo de desenvolvimento do projecto, no qual se podem identificar dois grandes momentos ao longo de cinco capítulos, onde cada um complementa os anteriores, sempre com foco nos objectivos propostos e na obtenção de resultados que reflitam os pensamentos e as percepções iniciais.

O capítulo introdutório procura caracterizar a importância das TI e SI na sociedade actual, reflectindo sobre os processos de construção de software e a importância destes no que diz respeito ao software criado nos dias de hoje, juntamente com a descrição dos tipos de sistemas normalmente utilizados para o ensino de SI.

No segundo capítulo, o primeiro grande momento, são estabelecidos os pontos principais que suportam todo o estudo realizado. É neste capítulo que é definido o problema, estudada a evolução do conhecimento na área dos DA e é feito o levantamento das oportunidades, problemas, desafios e complexidade do tratamento de DA com inclusão de nós de decisão e respectivos cenários.

Os dois capítulos que se seguem marcam o segundo grande momento deste trabalho. No capítulo três é explicada a abordagem feita à investigação proposta, é apresentado o método de investigação escolhido, bem como as técnicas de investigação que melhor se lhe adequam, assumindo assim o papel de explicar o que é feito para abordar o problema. O quarto capítulo complementa o terceiro, já que aí é apresentada a forma como o problema é tratado, descrevendo todo o processo de construção do sistema de apoio ao ensino de DA da UML, desde a recolha e análise de dados, passando pela modelação da ferramenta, até à concepção da solução.

Por fim, no quinto capítulo, é feita a revisão de todo o trabalho desenvolvido, são apresentadas as dificuldades encontradas, discutidos os resultados finais e identificados aqueles que pensamos serem os contributos do estudo para as TI e os SI. Neste capítulo, são apresentadas também as considerações finais do trabalho e identificado o trabalho futuro que poderá complementar o presente estudo.

2 Fundamentos da modelação em sistemas de informação

No capítulo que agora se inicia procura-se descrever de maneira clara e concisa, todo um conjunto de ideias sobre todos os temas que são abordados no presente estudo para que deste modo seja facilitada a compreensão do trabalho desenvolvido nos capítulos seguintes.

As organizações para serem bem sucedidas necessitam de um SI eficaz suportando desde os seus níveis operacionais até aos níveis estratégicos (Varajão, 2005). Para tal acontecer é necessário que estes sejam cuidadosamente concebidos, construídos e implementados, de modo a suportar convenientemente a organização (McKeown, et al., 1993).

Neste capítulo são abordados e realçados diversos aspectos cuja compreensão consideramos fundamental para a melhor percepção da modelação de SI, dando maior ênfase a uma das suas disciplinas, a Engenharia de Software (ES) e um dos tipos de diagramas usados nesta disciplina, os DA.

2.1 Engenharia de Software

Esta secção (2.1) apresenta conceitos da Unified Modeling Language, da ES, de DA e, mais propriamente, das decisões e dos cenários alternativos do DA.

Uma primeira definição dada à ES (Naur, et al., 1968) explica que esta é responsável pelo estabelecimento e uso dos princípios da engenharia de maneira a se obter software económico, fiável e que funcione eficientemente em máquinas reais.

Já mais recentemente, o IEEE Standard Glossary of Software Engineering Terminology (IEEE, 1991), definiu a ES como a aplicação de uma abordagem sistemática e disciplinada, quantificável ao desenvolvimento, operação e manutenção de software, isto é, a aplicação da engenharia para o software.

Sommerville (Sommerville, 2006) definiu ES como uma área de conhecimento composta por teorias, métodos e conjuntos de ferramentas necessários à geração de um produto de software.

De acordo com van Vliet (van Vliet, 2007), ES diz respeito aos métodos e técnicas para o desenvolvimento de grandes sistemas de software. O substantivo engenharia é usado para sublinhar uma aproximação sistemática ao desenvolvimento de sistemas que satisfaçam requisitos e limitações organizacionais.

Assim, mais de cinquenta anos passaram desde a primeira definição dada e ainda hoje não existe nenhuma definição universalmente aceite para a ES (Meyer, 2001).

Desta forma pode afirmar-se que a ES tem várias facetas. A ES não é certamente o mesmo que programação, embora a programação seja um ingrediente importante na ES. Também a matemática tem um papel importante no que à ES diz respeito, pois é necessário que o software tenha fórmulas correctas, assim como a psicologia e a sociologia são necessárias na medida em que é necessário ter em atenção a comunicação entre o homem e a máquina.

Assim, o termo ES pode levar-nos a uma analogia entre a construção de programas e a construção de casas, isto porque em ambos os casos trabalha-se um conjunto de funções usando técnicas científicas e de engenharia de uma maneira criativa, técnicas essas que foram e são aplicadas de uma maneira satisfatória na construção de objectos físicos e que são também bastante úteis quando aplicadas à construção de software (Eischen, 2002), nomeadamente o desenvolvimento do produto num determinado número de fases, um planeamento cuidado dessas mesmas fases e o contínuo acompanhamento em todo o processo.

2.2 A Unified Modeling Language

A UML surge em meados dos anos 90 através da combinação de diversos métodos de ES Orientados ao Objecto (OO) criados por competidores directos (Dobing, et al., 2006), sendo que o controlo da sua evolução foi colocado nas mãos do Object Management Group (www.omg.org) (OMG) e desta maneira tornou-se mundialmente aceite como padrão de modelação para o desenvolvimento de software OO.

Assim, alguns dos seus criadores (Rumbaugh, et al., 1999), definiram a UML como uma linguagem visual genérica de modelação que pode ser usada para especificar, visualizar, construir, e documentar os componentes de um sistema de software.

A UML é actualmente uma ferramenta largamente disseminada para a modelação de SI. No entanto, entre a esquematização conceptual de um sistema e a sua implementação existe um passo crucial de transformação que ainda não é conseguido, ou seja, a UML não é uma linguagem gráfica de programação (Gamito, et al., 2007) é sim, uma linguagem de modelação.

Na ES a UML afirmou-se como padrão para a especificação gráfica de aspectos estáticos e dinâmicos e em 2003 o OMG lançou a versão 2.0 da UML (Eichelberger, et al., 2003). Desta forma o número de diagramas diferentes cresceu de 9 para 13, sendo que novos tipos de diagramas foram introduzidos nesta versão enquanto alguns diagramas já existentes aumentaram em grande escala a sua complexidade.

Os 13 diagramas da UML estão descritos na Tabela 1. Alguns diagramas, como o diagrama de classes e o diagrama de estados, já figuravam nas especificações da UML desde as suas especificações iniciais, sendo que, por exemplo, o DA só viu reconhecida a sua importância aquando do lançamento da UML 2.0.

Tabela 1 - Descrição dos diagramas da UML

Diagrama	Descrição
Classes	Usado para modelar classes as suas classes, recursos e relações.
Actividades	Usado para modelar processos de negócios, fluxos de dados ou a lógica de um sistema.
Pacotes	Usado para modelar elementos em pacotes, bem como dependências entre pacotes.
Objectos	Usado para modelar os objectos e as suas relações num dado momento. Também conhecidos como diagramas de instâncias.
Estrutura Composta	Usado para modelar a estrutura interna de uma classe.
Componentes	Usado para modelar um conjunto de componentes e as suas inter-relações.
Deployment	Usado para modelar a arquitectura de execução dos sistemas. Isto inclui os nós, os ambientes de execução entre hardware e software, assim como o *middleware* que os liga.
Casos-de-uso	Usado para modelar casos-de-uso, actores e as suas inter-relações.
Estados	Usado para modelar os estados em que podem estar os objectos, assim como as transições entre estados.
Comunicação	Usado para modelar o fluxo de mensagens entre instâncias de classes.
Sequência	Usado para modelar a ordem pela qual as mensagens são trocadas entre as instâncias de classes.
Tempo	Usado para modelar as diferenças de estados de um objecto ao longo do tempo.
Interactividade	É uma variante dos diagramas de actividades, que mostra o fluxo de controlo dentro de um sistema ou processo de negócio.

Deste modo, quando se projecta algo de novo, torna-se conveniente recorrer a modelos que representem aquilo que irá ser desenvolvido. Esses modelos constituem uma representação abstracta de uma realidade projectada para o futuro (Mota, et al., 2004). A UML oferece uma forma padrão de criar esses modelos, permitindo a simplificação do processo de concepção de software, que por si só é bastante complexo, através do uso de uma forte componente gráfica, fazendo uso de imagens como elemento de comunicação e da utilização de um conjunto limitado de símbolos.

2.2.1 Diagramas de Actividades

O DA é um caso de persistência e sucesso que se foi construindo ao longo dos tempos. Na UML 1.X estes eram considerados e usados apenas em casos especiais enquanto desde o surgir da UML 2.0 os DA foram reconhecidos com parte integrante de toda uma grande família de técnicas da linguagem de modelação (Siebenhaller, et al., 2006).

As actividades e os correspondentes DA pertencem aos conceitos básicos da UML, que se tornou a linguagem padrão para especificar, visualizar e documentar sistemas de software (Siebenhaller, et al., 2006). A OMG (OMG, 2003) refere-se aos DA como diagramas com uma construção semelhante a fluxogramas com o intuito de exprimir sequência, escolha e execução paralela de actividades.

Também Sparx (Sparx, 2007) tem uma definição semelhante para os DA, e refere que estes são usados para mostrar a sequência de actividades, na medida em que mostram o fluxo de trabalho de um ponto inicial até ao ponto final, detalhando a maior parte dos caminhos de decisão existentes na progressão de eventos contidos na actividade, podendo ser usados também para detalhar situações onde o processamento paralelo possa ocorrer durante a execução de algumas actividades.

De igual forma Bhattacharjee (Bhattacharjee, et al., 2009) classificam os DA como um dos mais importantes artefactos usados na UML, pois estes são usados para modelar a sequência de acções como parte do fluxo de um processo de acções e seus resultados.

Assim, os DA são um dos diagramas da UML usados para modelar os aspectos dinâmicos de um sistema (Hammer, et al., 1998) e quando isto acontece, os DA são geralmente usados de duas maneiras (OMG, 1999):

- Para modelar um fluxo de trabalho, onde o foco se encontra nas actividades sob o ponto de vista dos actores que colaboram com o sistema;

- Para modelar uma operação, onde são usados como fluxogramas ou como detalhes de um sistema computacional (Chang, et al., 2001).

Já mais recentemente Eshuis (Eshuis, et al., 2001) definiu também quando se deverão ou não usar DA.

Deverão usar-se DA quando se pretende:

- Analisar um caso-de-uso;
- Descrever um algoritmo complicado de uma maneira sequencial;
- Perceber fluxos de trabalho.

Não deverão usar-se DA quando se pretende:

- Tentar visualizar como os objectos interagem. Neste caso um diagrama de interacção é mais simples e dá uma perspectiva mais clara das colaborações entre objectos;
- Tentar visualizar como um objecto se comporta durante o seu ciclo de vida. Para este efeito é preferível utilizar um diagrama de estados.

Desta forma, um dos grandes objectivos do DA é modelar o fluxo de processos de acções que são parte de uma grande actividade. Em projectos nos quais os casos-de-uso estejam presentes, os DA podem modelar um caso-de-uso específico com um maior nível de detalhe, sendo que podem ser também usados independentemente dos casos-de-uso para modelar processos de negócio (Lieberman, 2001).

Por causa do seu modelo de fluxo de processos, o DA foca-se na sequência de execução de acções e nas condições que fazem despoletar essas acções.

Assim, a actividade é despoletada por um ou mais eventos e uma actividade pode resultar em um ou mais eventos que por sua vez, podem despoletar outras actividades ou processos. Os eventos, que na UML são fluxos de mensagens, começam com um símbolo inicial e terminam com um símbolo final, havendo actividades no meio desses símbolos conectadas por outros eventos (Chang, et al., 2001).

Como todos os diagramas da UML, o propósito número um dos DA é comunicar a informação eficazmente. A razão principal para incluir um ou vários DA num modelo de sistema é devido ao facto destes modelarem o fluxo de processos das várias actividades (Bell, 2003) sendo que a sua maior virtude assenta no facto de estes suportarem e incentivarem o comportamento paralelo, ou actividades concorrentes, o que faz dos DA uma excelente ferramenta para modelar fluxos de trabalho.

Desta forma, e para concluir esta secção, pode considerar-se que, tal como todas as técnicas de modelação, também os DA têm as suas forças e fraquezas, sendo que revelam todas as suas virtudes quando combinadas com outros diagramas e outras técnicas.

2.2.1.1 A Sintaxe dos Diagramas de Actividades

A subsecção que se inicia agora tem um carácter conceptual e descritivo, de maneira a que, quem leia o presente estudo e se sinta de alguma forma não familiarizado com o conteúdo, perceba alguns dos conceitos dos DA, a sua semântica mais básica e os usos que se poderão dar-lhes.

Tal como foi escrito anteriormente, basicamente, um DA é usado para mostrar uma sequência de actividades. Os DA mostram o fluxo de trabalhos desde um ponto inicial até um ponto final, detalhando o mais possível os caminhos de decisão que ocorrem na progressão dos contidos na actividade (Sparx, 2007). Assim, os DA têm diversos constituintes, cada um com a sua função e a sua importância. De seguida descrevemos os constituintes dos DA que achamos serem fundamentais para uma boa percepção destes diagramas: a Actividade, as Transições ou Fluxos de Controlo, a Actividade Inicial ou Nó Inicial, a Actividade Final ou Nó Final, as Swimlanes ou Linhas de Responsabilidade, os Nós de Decisão e os Nós de Divergência/Convergência ou Barras de Sincronização.

Actividades

Uma actividade é a especificação da sequência de acções que são realizadas no decurso dessa mesma actividade. Para mostrar graficamente uma actividade é usado um rectângulo com cantos arredondados. Na Figura 1 é mostrado um exemplo de uma actividade.

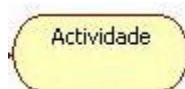



Figura 1 - Exemplo de uma Actividade dos Diagramas de Actividades

Transições ou Fluxos de Controlo

Uma transição ou fluxo de controlo permite descrever a sequência pela qual as actividades se realizam. A sua notação é uma seta. Na Figura 2 é mostrado um exemplo de um fluxo de controlo.

Figura 2 - Exemplo de uma Transição dos Diagramas de Actividades

Actividade Inicial ou Nó Inicial

Uma actividade inicial ou nó inicial pode ser uma actividade definida apenas para identificar o início do diagrama e é descrita por um círculo preto. Na Figura 3 é mostrado um exemplo de um nó inicial.

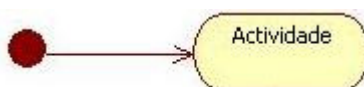



Figura 3 - Exemplo de um Nó Inicial dos Diagramas de Actividades

Actividade Final ou Nó Final

Tal como o nó inicial, o nó final pode ser uma actividade definida apenas para identificar o final do diagrama. Para se representar o nó final usa-se um círculo com outro círculo preto no seu interior. Na Figura 4 é mostrado um exemplo de um nó final.

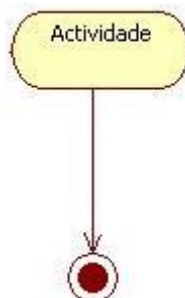



Figura 4 - Exemplo de um Nó Final dos Diagramas de Actividades

Swimlanes ou Linhas de Responsabilidade

As *swimlanes* são usadas para separar as actividades por quem é responsável por elas. Graficamente são mostradas como linhas, ou pistas, verticais ou horizontais. Na Figura 5 é mostrado um exemplo de uma *swimlane* vertical.

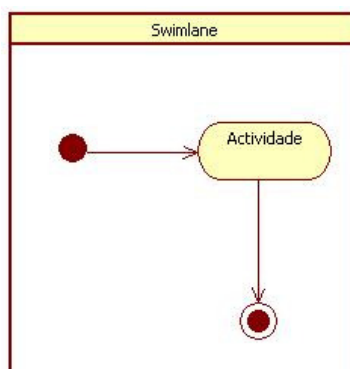



Figura 5 - Exemplo de uma *Swimlane* dos Diagramas de Actividades

Nós de Decisão

Os nós de decisão apresentam como notação a forma de um diamante e utilizam expressões booleanas entre parênteses rectos "[ ]" que servem de guarda para o prosseguimento do fluxo de trabalho do diagrama. Na Figura 6 é mostrado um exemplo de um nó de decisão (diamante mais condições de guarda).

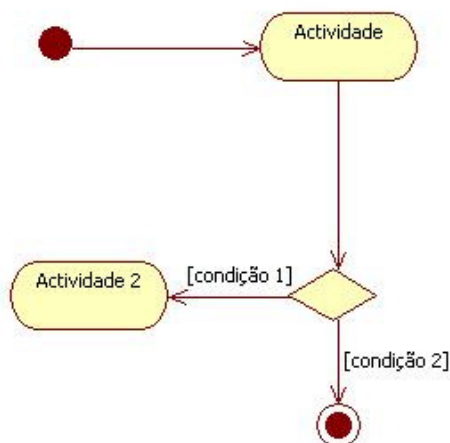



Figura 6 - Exemplo de um Nó de Decisão dos Diagramas de Actividades

Nós de Divergência/Convergência (barras de sincronização)

Os nós de divergência e convergência têm a mesma notação, duas barras horizontais ou verticais (dependendo da orientação do diagrama). Estas barras indicam o início e o fim de actividades concorrentes no decorrer do diagrama. Na Figura 7 é mostrado um exemplo de um nó de divergência e um nó de convergência.

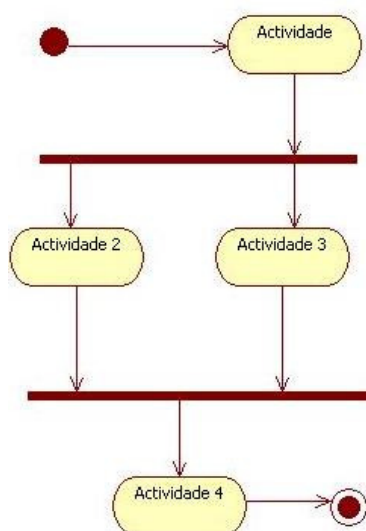



Figura 7 - Exemplo de Nós de Divergência/Convergência dos Diagramas de Actividades

Os DA apresentam muitos outros aspectos que tornariam este subcapítulo demasiado extenso e descritivo, por isso optou-se por colocar apenas os elementos essenciais que possibilitam uma compreensão razoável da ferramenta apresentada no capítulo 4.

A presente dissertação foca o processamento das decisões e dos cenários alternativos que podem existir num DA, de forma que este possa ser visualizado sob a forma de animação, simulando um caso real e facilitando assim o seu entendimento.

2.2.1.1.1 Decisões e cenários alternativos

Dependendo do processo de negócio representado num DA, por vezes pode surgir a necessidade de criar cenários alternativos de execução do processo, onde cada um resulta de uma condição de guarda que terá de ser verdadeira para que o cenário que se lhe sucede seja executado. Sempre que se verifique a necessidade de criar vários cenários, estes devem obrigatoriamente iniciar-se com a inclusão de um nó de decisão, também designado por ponto de decisão, ou simplesmente decisão.

Segundo Arlow (Arlow, et al., 2002), uma decisão representa cenários alternativos baseados em expressões de guarda booleanas. A sintaxe da UML para uma decisão é um símbolo em diamante. Só pode haver uma transição à entrada de uma decisão, podendo haver várias transições (cenários) à saída. Cada cenário de uma decisão é protegido por uma condição de guarda, sendo seguido esse cenário se, e só se, a sua condição de guarda for verdadeira.

A palavra-chave *else*[1] só pode ser utilizada numa das condições de guarda que represente uma saída do nó de decisão, não podendo ser combinada com as demais condições do mesmo nó. Assim, este termo serve para abreviar a disjunção de todas as demais condições de guarda de uma decisão (Eshuis, 2002).

De acordo com o anteriormente descrito, a correcta utilização das decisões e respectivas condições de guarda, obriga ao seguimento de apenas um cenário em cada execução de todo o processo de negócio. Esta regra tem duas consequências que é necessário entender: por um lado, a utilização de nós de decisão eleva o grau de complexidade dos DA, na medida em que será mais difícil entender que cenário será executado à saída da decisão; por outro lado, ao utilizarmos este componente na construção de DA, estamos a simplificar o processo de execução, já que apenas um cenário será percorrido de cada vez. No entanto, estas questões nunca devem ser motivo para decidir pela inclusão ou não de um nó de decisão num DA, cabendo apenas essa decisão, ao facto de existirem no processo duas ou mais hipóteses possíveis de execução, mas onde só uma pode ser tomada.

A complexidade de um DA também se deve ao número de nós de decisão nele incluídos, reflectindo-se no entendimento geral do processo de negócio, por parte do engenheiro de software e/ou programador. No que diz respeito à execução do processo por parte da máquina, essa complexidade não existe, já que esta decide por um cenário com base numa condição de guarda previamente definida e programada.

De qualquer modo, pode dizer-se que a aceitação dos DA como ferramenta de modelação de software e consequente inclusão dos nós de decisão nos diagramas, facilita a representação e compreensão dos mesmos por parte dos engenheiros, programadores e alunos de SI e ES, bastando para isso que estes tenham presente quais as condições em que tal componente deve ser utilizado e como se processa a execução dos DA com estas características.

2.3 Software de simulação

A modelação de sistemas recorrendo previamente à simulação tem sido largamente usada em áreas como a saúde, redes de comunicações, processos empresariais e comunicações, e a crescente criação de produtos mais elaborados requer uma maior flexibilidade e rentabilidade em processos produtivos assim como uma maior exigência na

[1] *Else*, em Português "senão", é o termo inglês utilizado pela maioria das linguagens de programação, pela algoritmia e também pela UML para definir o percurso que deve ser seguido pela execução de um programa de computador quando as condições especificadas não contemplam todas as opções.

facilidade de manutenção dos sistemas. Assim surgiu o software de simulação, programas de software onde o utilizador assume um papel interactivo permitindo-lhe experimentar e controlar um acontecimento como se da realidade se tratasse.

Devido ao aumento das pressões competitivas, muitas empresas foram forçadas a implementar eficiência e ao mesmo tempo reduzir custos. A modelação com ajuda de software de simulação tornou-se algo popular, (Doukidis, et al., 1990) com uma ampla gama de aplicações (Gogg, et al., 1993) usada para encontrar estratégias para o melhoramento do desempenho e para criar processos alternativos aos já utilizados.

À medida que os avanços tecnológicos vão acontecendo, o software de simulação dá a oportunidade às empresas e ao sistema educativo de melhorarem os seus processos e de melhorarem as capacidades das pessoas que as integram.

Os sistemas de produção das grandes empresas possuem ambientes extremamente complexos, com um grande número de variáveis que afectam o seu desempenho. Isto faz com que o treino apropriado à gestão destas se torne mais difícil de dia para dia e requeira bons conhecimentos de conceitos mas também alguma experimentação (Chiavenato, 1990). Assim o software de simulação torna-se um bom ajudante, na medida em que proporciona um ambiente capaz de transmitir conhecimentos, inclusivamente de carácter prático e propiciam a aprendizagem com maior participação e aproveitamento.

O número de empresas que usam software de simulação para minimizar problemas de produção e administração tem crescido rápido e acentuadamente, pois os seus administradores e gestores já compreendem os benefícios que o uso desta técnica possibilita levando assim a que estes a incorporem nas suas operações diárias (Gramigna, 1993).

Deste modo, podemos concluir que existem vantagens e desvantagens no uso deste tipo de software. Axelrod (Axelrod, 1997) identifica as seguintes vantagens dos pacotes de software de simulação: possibilidade de representar e avaliar sistemas reais ou sem existência física; redução de custos associados ao desenvolvimento de protótipos físicos, possibilidade de efectuar análises do tipo “E se...?”; avaliação detalhada do sistema simulado; identificação das variáveis com um maior impacto sobre o desempenho de um sistema; e possibilidade de visualização dinâmica (em 2D ou 3D) do sistema simulado.

Como desvantagens podem referir-se os elevados custos deste tipo de software, um consumo exagerado de tempo para o desenvolvimento dos modelos, necessidade de treino especializado para manipular o software e a dificuldade de interpretação dos resultados finais, pois a simulação apenas fornece estimativas dos parâmetros do sistema.

Existem características que os utilizadores também gostariam de ver implementadas, tais como maior flexibilidade e melhor compatibilidade entre aplicações de software. No entanto não existem pacotes de software que possam implementar todas as características que os utilizadores pretendem ver implementadas. Nenhum sistema existente consegue ser ao mesmo tempo fácil de usar e de aprender, ter um baixo custo, ter capacidades gráficas excelentes e grande flexibilidade (Hlupic, 2000).

No que diz respeito ao sistema de aprendizagem, o software de simulação garante ao aprendiz que o modelo simulado existe; todo um cenário pode ser analisado visualmente desde o início de uma qualquer produção até ao seu fim; e pode ser analisado ao pormenor conhecendo todos os aspectos do desenvolvimento.

Assim, existe a preocupação de tentar criar software de simulação que recrie o mais próximo possível a realidade e cabe aos criadores de software criar produtos que permitam ao aprendiz conhecer o sistema a simular e agir directamente sobre ele, pois este software propicia uma aprendizagem com maior participação e aproveitamento.

O que se irá passar com a evolução da tecnologia de simulação é difícil de prever, no entanto é certo que o software de simulação do futuro terá de ter em conta as necessidades dos utilizadores actuais e as novas necessidades que continuam a surgir. Banks (Banks, 1994) argumenta que o futuro do software de simulação será mais uma contínua evolução e melhoramento do que revolução. Uma das razões apontadas é a de que o software de simulação tem um legado de clientes que necessitam de apoio com novas funcionalidades e melhoramentos. Os criadores de software de simulação necessitam de equilibrar as actuais necessidades dos clientes com as necessidades do mercado.

Como conclusão deste subcapítulo, podemos referir que, o software de simulação começa a surgir em força entre nós pois os seus utilizadores já consideram este tipo de software de fácil uso e com boas qualidades em termos de interface, não obstante também lhe serem apontados inconvenientes, tais como o preço e a limitação na simulação de situações complexas (Hlupic, 1998).

2.4 Sistemas de software aplicados ao ensino

É evidente que as TI têm vindo a alterar os métodos, finalidades e a percepção do potencial da educação (Furneaux, 2004). Os avanços na ciência e tecnologia têm forçado algumas mudanças nos métodos de ensino. Hoje em dia o uso do computador como meio para ensinar ou como um instrumento de aprendizagem é tão vulgar que quase passa despercebido.

O software educativo, resultado de um processo de desenvolvimento de 30 anos, são ferramentas populares, úteis e necessárias para a educação de hoje e provavelmente continuarão a sê-lo no futuro (Monadjemi, et al., 2001), pois as enormes capacidades de processamento de informação do computador fazem com que seja possível utilizá-lo para adaptar as rotinas do ensino às necessidades e ao desempenho de cada aluno (Suppes, 1966).

Oliveira (Oliveira, et al., 2001) enquadra o software aplicado ao ensino em duas categorias: software de aplicação e software educativo.

Na categoria de software de aplicação entra o software que não foi desenvolvido com finalidades educativas, mas que pode ser utilizado para esse fim. São os programas de uso geral no mercado e utilizados em contexto de ensino como, por exemplo, os programas de bases de dados, processadores de texto, folhas de cálculo e editores gráficos. Segundo Carvalho (Carvalho, et al., 2003) o software de aplicação pode também ser usado para construir um software aplicado ao ensino através, por exemplo, da programação de folhas de cálculo que armazenam e executam equações de uma modelação de um sistema real.

O objectivo destes programas é favorecer os processos de ensino-aprendizagem e são desenvolvidos especialmente para construir o conhecimento relativo a um conteúdo didáctico, que possibilita a construção do conhecimento numa determinada área com ou sem a mediação de um professor (Jucá, 2006).

Segundo Taylor (Taylor, 1980), existem três tipos de software aplicado ao ensino e podem ser designados como “Tutor”, “Ferramenta” ou “Tutelado”.

Como “Tutor”, o software instrui o aluno, desempenhando praticamente o papel do professor.

Como “Ferramenta” os alunos aprendem a usar o computador para adquirir e manipular informações, utilizando muitas vezes software de uso genérico de outras áreas, como: processadores de texto, folhas de cálculo, bases de dados, etc.

Já na forma de “Tutelado” seria classificado o software que permite ao aluno “ensinar” o computador. São disso exemplo as linguagens de programação como Pascal e LOGO, onde a linguagem em si não é usada como objecto de estudo mas serve como um canal para a representação das ideias.

Existem outros autores que preferem classificar o software de acordo com a maneira como ele manipula o conhecimento: geração de conhecimento, disseminação de conhecimento e gestão da informação (Knezek, et al., 1988).

Deste modo é visível que os recentes avanços na ciência e na tecnologia levaram mudanças nos métodos de ensino e, desta forma surgiu também o software aplicado ao ensino. As facilidades multimédia, os computadores mais rápidos e mais baratos e a WWW caracterizam estas ferramentas poderosas e disponíveis a toda a hora (Monadjemi, et al., 2001).

No entanto este processo está longe de estar concluído. Vários autores (Oblinger, 1997) (Hulme, et al., 2003) consideram que as instituições de ensino que se recusem a adoptar e integrar as TI em todos os níveis dessa mesma instituição sofrerão graves consequências.

Também a maior parte do software aplicado ao ensino não está ainda adaptado às necessidades dos utilizadores. De facto há uma grande distância entre os protótipos do software aplicado ao ensino adaptado às necessidades dos utilizadores e entre o software aplicado ao ensino comerciais (já em uso) (Boticario, et al., 2006).

Desta forma, é necessário que se perceba que o computador e o software aplicado ao ensino são possibilidades de representação e exposição da informação e tornando-se assim uma alternativa ou estratégia de aprendizagem para uma melhor compreensão da realidade.

Entretanto, é importante não esquecer que um progresso relacionado com um critério específico pode manifestar-se de diferentes formas, em diferentes alunos e uma mesma acção pode, para um aluno, indicar avanço em relação a um critério estabelecido e para outro, não. Por isso, além de serem necessários indicadores precisos, os critérios de avaliação devem ter em conta todo o seu conjunto, considerados de forma contextual e analisados à luz dos objectivos que realmente orientaram o ensino oferecido aos alunos. Se o propósito é avaliar o processo e o produto, não há nenhum instrumento de avaliação da aprendizagem melhor do que procurar identificar porque é que o aluno teria dado as respostas que deu às situações que lhe foram propostas (Papert, 1997).

Por fim, para a avaliação de um software aplicado ao ensino é preciso ter em consideração a utilidade dessa ferramenta no processo de aprendizagem do aluno e na construção dos seus conhecimentos.

Portanto, ao avaliar e escolher um software aplicado ao ensino deve-se ter em conta o perfil do aluno e a proposta pedagógica da instituição de ensino na qual será desenvolvido o trabalho e ainda considerarem-se aspectos como a clareza e objectivo do software (Pinto, 2003).

2.5 O ensino em Engenharia de Software

Ainda antes da entrada no novo milénio, os sistemas de software tornaram-se parte essencial de todas as actividades diárias e dos negócios na economia global. A qualidade deste software depende de uma adequada formação e permanente actualização de conhecimentos dos criadores de software.

Hoje em dia, os criadores de software são ensinados de um modo tradicional, mas infelizmente, este modo não criou a quantidade e qualidade necessárias de criadores, não chegando para satisfazer a crescente procura (Shaw, 2000).

Na indústria de software de hoje, a um engenheiro de software não só é expectável que conheça e lide com várias técnicas e desafios, mas também que lide com questões não técnicas que surjam de situações difíceis que apareçam em projectos. Estas questões incluem geralmente a percepção dos requisitos do cliente, trabalhar em equipa, organizar a divisão do trabalho e o lidar com a pressão do tempo e prazos de entrega. Por isso, ensinar ES não requer apenas o estudo teórico mas também disponibilizar aos alunos a experiência de questões não técnicas que surjam em projectos de software.

O desenvolvimento de sistemas de software apresenta sobretudo problemas de ordem não técnica, devido ao facto do planeamento e análise serem essenciais. Relendo a bibliografia sobre o desenvolvimento de sistemas, é fácil de constatar que o pobre planeamento de sistemas é a maior causa dos projectos mal sucedidos (Metzger, et al., 1996), sendo a segunda causa a insuficiente percepção dos requisitos pretendidos pelo cliente (Jackson, et al., 1995). A comunicação e a gestão são muito importantes quando se trata de projectos de software. Assim, gerir um projecto de software significa, primeiro que tudo, recolher o máximo de informação possível, mantendo o grupo de trabalho bem informado e coordenar as necessidades individuais e tarefas para assim se atingir os objectivos pretendidos.

Existem vários livros de ES que descrevem as dificuldades e desafios da aprendizagem de ES, alguns deles (Brugge, et al., 2001) focam-se mais em aspectos de design enquanto outros (Collofello, et al., 1998) colocam o foco nas questões da gestão de projectos. No entanto, a melhor maneira de realmente preparar os alunos para o desenvolvimento de projectos bem sucedidos é colocá-los numa situação de “aprender fazendo”. Numa perspectiva das universidades, encontrar um conceito de ensino que disponibilize aos alunos um conhecimento prático, é ainda um desafio (Gnatz, et al., 2003).

Num curso típico onde exista a disciplina de ES, o seu ensino poderá falhar, na medida em que estes não conseguem as capacidades necessárias para o desenvolvimento de soluções

quando saem para o mundo do trabalho, pois existe uma grande diferença nas capacidades de ES adquiridas na universidade e as capacidades que são necessárias para um engenheiro de software na organização de um projecto típico (Callahan, et al., 2002). Este problema parece ter origem na maneira como a ES é geralmente ensinada: a teoria é apresentada na forma de leitura e é posta em prática (limitada) em projectos de grupo.

Um projecto de ES melhora as capacidades dos alunos para o desenvolvimento profissional de software. Os alunos necessitam de experimentar o *stress* e a tensão de trabalhar em grupo, atribuir e aceitar responsabilidades, lidar com diferentes grupos (internos e externos, por exemplo clientes) e terem normas e regras que rejam o seu trabalho e apenas experimentam isto quando lidam e trabalham no desenvolvimento de um sistema dado por um cliente que é externo ao seu curso (Lethbridge, 2000).

Embora tanto as leituras da teoria como os projectos de grupo sejam essenciais para a aprendizagem, existem falhas profundas em todo o processo da ES propriamente dita, pois a leitura da teoria, que apenas permite a aprendizagem passiva, e a dimensão (geralmente reduzida) dos projectos de grupo, são técnicas demasiadamente limitadas, não contendo muitas das características fundamentais da ES do mundo real (Navarro, et al., 2003).

Assim, a melhor (e provavelmente única) motivação para a aprendizagem de gestão de projectos é terem as experiências de projectos falhados devido à insuficiente descrição e detalhe na especificação inicial destes. Desta maneira, depois de verificarem que falharam como gestores de projecto, os alunos começarão a fazer perguntas e a ouvir as respostas.

No entanto, não existem universidades, nem tempo, nem currículo de curso que permita um número suficiente de projectos de maneira a que cada aluno possa ser um gestor de projectos (Mingins, et al., 1999). Contudo, a gestão de projectos é algo que é feito com algum rigor em empresas, isto é, quando um aluno passa da universidade para o mundo do trabalho; por isso o treino que estes alunos têm quando passam esta fase é algo que se pode confiar em parte para colmatar o que falhou nos seus anos de formação.

Mas, nem sempre é assim e os bons resultados são raros, ou seja, o que não é ensinado nas universidades ou escolas, provavelmente nunca ficará bem presente na cabeça dos futuros profissionais da área, o que faz com que haja uma pobre gestão de projectos de software.

A chave para uma aprendizagem bem sucedida é a motivação. Se numa acção pedagógica se mencionar uma qualquer ferramenta de software livre, os alunos ouvi-la-ão com atenção e aplicarão o conhecimento adquirido imediatamente. Se se falar em gestão de projectos, os alunos mal ouvirão o que será dito, isto porque não consideram que isso seja um

problema real nem imaginam a maneira de como aplicar a gestão de projectos nos pequenos projectos que vão desenvolvendo ao longo do seu percurso escolar (Claypool, et al., 2005).

Deste modo, a simulação, frequentemente usada para o treino em várias áreas de estudo, como a economia e a aviação (Claypool, et al., 2005), pode ser usado no que à ES diz respeito.

Pode-se, neste caso, fazer uma analogia interessante com os pilotos de avião ao aprenderem a sua profissão. Estes precisam de saber como se comportar em situações perigosas, mas ninguém os sujeitará a tamanho risco apenas por uma questão de treino. Assim, as empresas de aviação disponibilizam simuladores poderosos que modelam todo o avião e o seu ambiente excepto para o piloto, que ao usar estes simuladores poderá experienciar todas as dificuldades sem qualquer risco e ainda com baixo custo (Collofello, et al., 1998).

A ES ainda não é uma profissão madura. O fosso existente entre o que é aprendido nos cursos que contenham esta disciplina e o que é necessário nas empresas ou instituições que contratam este tipo de serviços é muito maior quando comparado com outras disciplinas da área da informática (Surendran, et al., 2000). Para a melhoria desta situação, o ideal seria incluir os alunos em projectos reais com prazos fixados. Contudo, esta situação seria impensável tendo em conta os custos inerentes e os prazos para a conclusão de projectos (geralmente maiores que o tempo de duração da disciplina). Deste modo, a solução passa pelo mesmo método já existente, ou seja, incluir os alunos em mini-projectos de grupo delimitando prazos de entrega e apostando na formação dos mesmos através da teoria.

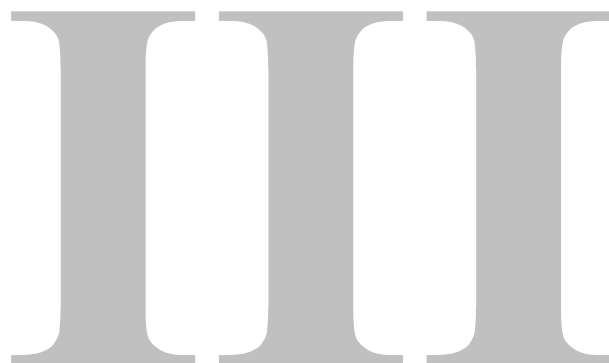
Para uma melhor compreensão dos diagramas da UML leccionados nesta disciplina, apresentamos nos capítulos seguintes um sistema de software para complemento ao ensino de DA através da simulação dos mesmos.

2.6 Problemas e desafios do projecto

Um dos principais desafios do desenvolvimento de software de simulação direccionado ao ensino é disponibilizar interfaces amigáveis, já que estes podem não estar familiarizados com este tipo de ferramentas e criar condições para que estas sejam interactivas o suficiente para permitir que os alunos experimentem e testem o problema que tentam resolver.

No caso específico de ferramentas de apoio ao ensino de DA, um desafio relevante poderá ser desenvolver uma ferramenta de simulação que, para além de permitir representar DA segundo as regras da UML, também permita a sua animação/simulação permitindo que o

utilizador relacione o problema apresentado com um cenário real e assim verificar se se trata de um problema de solução lógica.

O problema que se coloca é a verificação da utilidade da ferramenta, ou seja, se na prática a solução responde às necessidades dos utilizadores e se esta se torna útil e aplicável em casos de estudo mais complexos. Para isso, normalmente recorre-se a experiências que validam as características e funcionalidades disponibilizadas. Essa verificação terá de ser realizada num universo onde estejam representados os mais variados perfis de utilizadores de modo a tirar partido da ferramenta desenvolvida, evitando assim que esta fique obsoleta.

3 Processo de investigação

Após os dois capítulos anteriores, de introdução e de fundamentação ao tema do presente estudo, pensamos ser importante e imprescindível um capítulo em que sejam clarificadas questões relacionadas com a ciência e com métodos de investigação.

A opção pela colocação deste capítulo depois do capítulo de fundamentação e antes do capítulo de apresentação da solução foi feita com o propósito de permitir que os diversos assuntos apresentados possam ser estudados de acordo com a sua fundamentação e dos processos que conduziram aos resultados obtidos durante e no final da método seguido.

Deste modo, numa primeira parte deste capítulo, é feita uma introdução sobre as razões da existência da ciência e a sua relação com a tecnologia. Numa segunda parte é caracterizado e explicado o tipo de abordagem definido para o desenvolvimento e condução deste projecto.

3.1 A ciência e a tecnologia

A linha que nos dias de hoje delimita a ciência e a tecnologia é muito vaga. Bunge (Bunge, 1980) delimita estas duas vertentes colocando a tecnologia como sendo a ciência aplicada, ou seja, a tecnologia poderá ser considerada como que um “filho” da ciência, quer seja para aplicar conhecimentos em pesquisas, produzir instrumentos úteis ou mesmo para obter lucros.

De acordo com Silva (Silva, 1996), a ciência pode ser dividida em duas: a ciência pura ou básica e a ciência aplicada ou tecnológica. A ciência pura ou básica seja ela teórica ou experimental, tem como único objectivo o enriquecimento do conhecimento humano. Para a ciência aplicada ou tecnológica, a ciência pura ou básica é um meio e não um fim, na medida em que a ciência aplicada pode ser vista como "o conjunto das aplicações da ciência básica" (Bunge, 1980).

De maneira semelhante, Galli (Galli, 1993) coloca a tecnologia no universo da história e da cultura, como aplicação da ciência para uso da sociedade, na medida em que esta faz a ponte entre as descobertas científicas e as suas leis para o uso da sociedade na produção e utilização de instrumentos ou técnicas.

Deste modo, a definição de tecnologia pode ainda incluir as capacidades de perceber, compreender, criar, adaptar, organizar e produzir instrumentos, produtos e serviços, onde a inovação e adaptação constituem a sua dinâmica. Já o cunho intelectual da ciência serve-se da abstracção teórica, do raciocínio lógico-matemático sobre um qualquer objecto ou fenómeno (Silva, 1996).

Assim, a Ciência, em sentido lato, pode considerar-se um "espaço" onde se acumulam verdades, se testam teorias, modelos, se procuram excepções às leis, onde se fazem e desfazem paradigmas, se revoluciona a matéria, a antimatéria e o espírito; um espaço onde o próprio conceito de Ciência não escapa à mudança dos tempos (Gonçalves, 1997).

3.1.1 Técnicas e métodos de investigação

Os métodos de investigação podem catalogar-se segundo diferentes ópticas. A perspectiva mais utilizada é a classificação em métodos quantitativos e qualitativos.

Os métodos que se apoiam em modelação matemática e em experiências laboratoriais são catalogados como métodos quantitativos e são os métodos mais utilizados na investigação em ciências naturais e nas engenharias. Os métodos qualitativos emergiram na investigação nas ciências sociais e têm como finalidade o potenciar da análise das pessoas e a sua adaptação ao meio que as envolve (Myers, 1997).

Quer seja um método de investigação quantitativo, quer seja qualitativo, este guiar-se-á sempre por uma abordagem filosófica que define os princípios metodológicos, epistemológicos e ontológicos em que o método se baseia (Orlikowski, et al., 1991).

Os vários métodos de pesquisa e investigação mais utilizados no âmbito da Engenharia de SI (Orlikowski, et al., 1991) apresentados na Tabela 2 explicitam um conjunto de técnicas e ferramentas passíveis de conduzir um processo de investigação. Cada um deles traduz uma das correntes filosóficas acima introduzidas.

Tabela 2 - Métodos de investigação em Sistemas de Informação

Método	Descrição
Estudo de Caso	Baseia-se na investigação de um facto actual no contexto de uma situação real numa organização. Apresenta como desvantagem o facto de se confinar a uma única organização, dificultando deste modo a generalização dos resultados obtidos. Este método insere-se na corrente filosófica do Positivismo e Interpretivismo.
Survey	Recolha de dados através de questionários ou entrevistas sobre o tema em estudo e a posterior análise dos dados recolhidos. Permite apurar dados sobre fenómenos do mundo real. Insere-se no Positivismo.
Etnografia	Método com raízes na Antropologia. Baseia-se no estudo de um objecto por experiência da realidade onde esse objecto se inclui. Insere-se no Interpretivismo.
Experiência de Campo	Uso de métodos experimentais em situações reais. Apresenta como vantagem o facto de trabalhar directamente com o objecto em estudo e como desvantagem o facto de ser difícil determinar-se exactamente todas as variáveis, de modo a ser possível repetir as experiências com exactidão. Insere-se na corrente filosófica do Positivismo.
Investigação-Acção	Neste método, o investigador é um elemento activo na realização do mesmo. A Investigação-Acção centra-se na análise de resultados obtidos por alterações impelidas no objecto em estudo. Insere-se no Interpretivismo.
Simulação	Centra-se na simulação do comportamento do processo em análise, permitindo deste modo a produção de vários cenários para análise de um determinado facto. Insere-se no Positivismo e no Interpretivismo.

Existem diferentes correntes filosóficas mas as que mais se destacam são a corrente Positivista e a corrente Interpretivista. A corrente Positivista arroga que a realidade é objectiva e independente de quem a observa e advoga ainda que a forma exacta de gerar

conhecimento é através da construção de teorias que são posteriormente legitimadas, recorrendo-se a testes organizados onde se verificam hipóteses. Por outro lado, a corrente Interpretivista diz que a realidade é o resultado da interpretação do observador, ou seja, o conhecimento da realidade só pode ser ganho através da compreensão e interpretação dos fenómenos em estudo, inserindo o observador no centro da realidade em estudo (Myers, 1997).

De maneira a aplicarem-se os métodos de investigação descritos é necessário obedecer a um conjunto de etapas. A definição destas mesmas etapas não é a mesma em todos os métodos, no entanto todos os projectos de investigação incluem três etapas chave.

Tabela 3 - Técnicas de recolha de dados

Técnica	Descrição
Análise de documentos	Baseia-se na recolha, leitura e análise de documentos escritos ou outros artefactos sobre a área de em estudo. Usa-se nos métodos Estudo de Caso, Etnografia e Investigação-Acção.
Elaboração de entrevistas	A elaboração de entrevista tem como objectivo o aprofundar de um determinado tema ou apreender o parecer de um determinado interveniente do facto em estudo. Usada nos métodos Estudo de Caso, Etnografia, Investigação-Acção e Estudos de Mercado.
Experiências	As experiências são elaboradas de maneira a atestarem a hipótese formulada por correlação das variáveis que caracterizam o facto em estudo. Usada no método de Experiências de Campo e na Investigação-Acção.
Observação	Esta técnica centra-se na observação de um conjunto de factos com o propósito de recolher dados sobre o que um conjunto de intervenientes faz. O observador pode ter um papel activo ou não dependendo do método de investigação usado. Utilizada nos métodos Estudo de Caso, Etnografia, Investigação-Acção e Estudos de Mercado.
Questionários	Elaboração de um misto de questões relacionadas com o tema da investigação. Técnica utilizada nos métodos Estudo de Mercado e Estudo de Caso.

Deste modo, o primeiro passo a ser dado num projecto de investigação é a definição do propósito do tema e o tipo de estratégia a seguir, seleccionando-se assim o método de investigação e a técnica a aplicar. O segundo passo a realizar será a recolha de dados utilizando-se uma ou mais técnicas de recolha de dados (isoladamente ou em conjunto), que se encontram descritas na Tabela 3. Por fim, uma última etapa que não poderá faltar num projecto de investigação é a análise dos dados recolhidos. Nesta última etapa, a análise de dados surge com o objectivo de conduzir o investigador na tarefa de verificar a similitude ou a diferença existente entre dados e na explicação das razões para os factos que ocorram.

A discriminação do pressuposto da investigação é um dos passos mais importantes de todo o processo de investigação e implica a realização e especificação das três fases descritas no parágrafo anterior.

Assim, o presente estudo encontra-se assente em três premissas iniciais:

- Área de Estudo: Sistemas de Informação.
- Elemento de Estudo: Diagramas de Actividades.
- Foco: Decisões e cenários alternativos.

O facto de existirem dúvidas a serem esclarecidas, relacionadas com níveis de percepção dos alunos na aplicação a desenvolver, a escolha do tipo de investigação recaiu no método de Investigação-Acção.

3.1.2 O método Investigação-Acção

A Investigação-Acção pode ser descrita como um método de pesquisa que se baseia em razões positivistas e que se centra na acção como um intento de mudança e na investigação um procedimento de compreensão (Hugon, et al., 1988).

De uma forma mais simples, pode também classificar-se este método como o "aprender fazendo", ou seja, o investigador identifica o problema, define métodos para o resolver, verifica se os métodos são efectivos e, caso contrário, define um novo método de acção.

O método da Investigação-Acção varia ciclicamente entre a acção e a reflexão que, de maneira continuada, aprimora métodos na recolha de informação e na interpretação que se vai criando de maneira a compreender-se determinada situação (Baskerville, 1999).

Este método segue uma série de premissas que permitem diferenciá-lo de uma simples resolução de problemas (Cohen, et al., 2000):

- É assumido o compromisso por parte do investigador de estudar um sistema e introduzir modificações no mesmo de maneira a seguir a direcção desejada;
- Combina o diagnóstico de cada uma das suas iterações com a reflexão, focando-se em problemas reconhecidos pelos participantes como discutíveis mas passíveis de serem modificados;
- Implica o envolvimento do investigador e dos participantes, sendo estes últimos considerados co-investigadores, uma vez que todos os pareceres são tidos em consideração.

Assim, para a realização um processo de investigação baseado no método de Investigação-Acção é necessário seguir quatro fases (Pérez Serrano, 1994):

1. Diagnosticar ou descobrir o "problema";
2. Construção do plano de execução;
3. Proposta prática do plano e observação de como este funciona;
4. Reflexão, interpretação e assimilação dos resultados. Replanificação.

De acordo com Kuhne (Kuhne, et al., 1997), as fases deste método adoptam a forma apresentada na Figura 8.

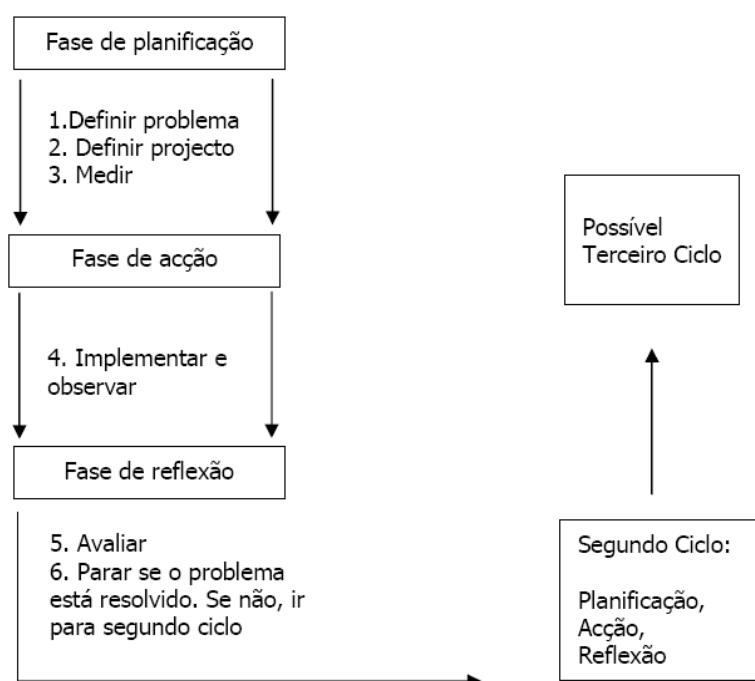



Figura 8 - Fases do método de Investigação-Acção

Fonte: (Almeida, 2001)

O ciclo da Investigação-Acção é forçosamente infindável, tendo sempre a acção revezando com a reflexão, em espiral ou em ciclos dentro de ciclos. No entanto, esta situação sempre incompleta, permite actuar com flexibilidade e assim melhorar para enfrentar a complexidade de sistemas de actividade humana. Os ciclos obrigatórios deste método obrigam a que a Investigação-Acção seja flexível mas ao mesmo tempo extremamente rigorosa, cada ciclo da acção implicando uma reflexão crítica, onde cada ciclo representa um novo planeamento e uma sequente acção (Dick, 2002).

3.2 Abordagem ao problema

De maneira a aplicar o método de investigação descrito anteriormente foi pensada como técnica de recolha de dados a técnica das experiências. A técnica das experiências é muito usada e tem uma grande importância, pois coloca o seu foco na observação da actividade científica "*in situ*", onde o investigador assume o papel de participante na situação que está a observar (Woolgar, 1982). Juntando a técnica das experiências com a técnica das entrevistas, onde o investigador questiona os participantes usando uma série de perguntas programadas previamente, é possível acercar os requisitos que os participantes destas experiências/entrevistas consideram ser os mais importantes para uma boa percepção do fenómeno em estudo.

3.2.1 Experiências realizadas

Todas as experiências realizadas no âmbito do presente trabalho foram elaboradas de forma independente e sequencial, tendo sido cada uma delas estruturada de forma a avaliar uma funcionalidade pensada para a simulação dos DA e assim seguir o comportamento cíclico do método Investigação-Acção na medida em que cada uma das experiências via a sua estrutura afinada de acordo com o resultado da experiência antecedente de modo a obter resultados apropriados às necessidades dos utilizadores.

De salientar ainda que todas as experiências tiveram como participantes alunos do primeiro ano da Licenciatura em Informática e da Licenciatura em Tecnologias da Informação e Comunicação da Universidade de Trás-os-Montes e Alto Douro. Como pré-requisito principal para a realização destas experiências era necessário que estes alunos nunca tivessem tido qualquer contacto com os diagramas da UML.

Para garantirmos que cada uma das experiências era realizada de acordo com o planeado, foram sendo criadas as duas tabelas representadas seguidamente. Na Tabela 4 encontra-se representada a sequência das fases de cada uma das experiências e a Tabela 5 lista cada um dos símbolos que representa uma fase e descreve seu o significado.

Tabela 4 - Sequência das fases das experiências

Experiência	Sequência
Pré-experiência	
1	
2	
3	
4	
5	
6	
7	
8	

Tabela 5 - Descrição das fases que se realizaram no decorrer das experiências

Símbolo	Descrição
	Disponibilizar ao aluno um diagrama de actividades exemplo.
	Mostrar a animação sem som ao aluno com cenários juntos e separados e com *fade-out* total e parcial das actividades.
	Mostrar a animação com som ao aluno com cenários juntos e separados e com *fade-out* total e parcial das actividades.
	Mostrar a animação sem som ao aluno com cenários juntos e com *fade-out* parcial das actividades.
	Mostrar a animação com som ao aluno com cenários juntos e com *fade-out* parcial das actividades.
	Mostrar a animação com som ao aluno com cenários separados e com *fade-out* parcial das actividades.
	Mostrar a animação com som ao aluno com cenários separados e com *fade-out* total das actividades.
	Pedir ao aluno para explicar o que percebeu daquilo que viu.
	Disponibilizar ao aluno um manual sintético sobre diagramas de actividades.
	Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Foi preparada uma pré-experiência para submeter todos os alunos participantes na experiência a todos os cenários possíveis, avaliando deste modo a reacção de cada um a situações desconhecidas. Esta experiência teve também como objectivo a preparação das experiências seguintes, tendo em conta o seu resultado em relação a tempos de resolução de exercícios e material de suporte para as experiências subsequentes.

A experiência número um teve como principal objectivo a avaliação em relação à presença do som, isto é, se os alunos participantes na experiência percepcionavam melhor o

problema existindo som de descrição em relação ao que estavam a visualizar ou se, por outro lado, a presença do som poderia ser considerada prejudicial.

A segunda experiência foi criada para verificar se o uso de cenários separados a partir de um ponto de decisão era mais benéfico para a melhor percepção do diagrama ao invés do uso de cenários juntos a partir de um ponto de decisão.

O objectivo da terceira experiência foi verificar se o uso de *fade-out* das actividades à medida que iam sendo mostradas, contribui para a compreensão deste tipo de diagramas. Nesta experiência foi testado o uso do *fade-out* total[1] e do *fade-out* parcial[2] das actividades.

A experiência número quarto apresentava como principal objectivo a avaliação das dificuldades que surgiam ao aluno quando resolvia o exercício proposto sem recorrer à visualização de DA exemplo nem à visualização da simulação de DA, tendo apenas como a ajuda o manual sintético.

Na quinta experiência a avaliação do desempenho dos alunos foi realizada mostrando apenas o DA exemplo e seguidamente foi pedido que os mesmos resolvessem o problema proposto sem recorrer à visualização da simulação de diagramas de actividade de modo que se pudesse percepcionar o modo como a simulação ajuda a uma melhor compreensão. Desta forma, os resultados desta experiência foram comparados com os resultados da sexta experiência. A sexta experiência apresenta-se como o oposto da quinta, pois foi pedido aos alunos que resolvessem o problema proposto tendo apenas visualizado o manual sintético e a simulação dos DA.

A sétima experiência foi usada para verificar se a visualização da simulação era benéfica para uma melhor percepção do problema após a visualização do DA exemplo e a oitava e última experiência foi criada para perceber se a visualização do DA exemplo após a visualização da simulação levaria à obtenção de melhores resultados na resolução do exercício final destas duas experiências.

---

[1] Com *fade-out* total referimo-nos ao desaparecimento completo da imagem que representa a actividade.

[2] Com *fade-out* parcial referimo-nos a um esbatimento da imagem, sendo que esta continuava visível ao participante.

3.2.2 Instrumentos de apoio às experiências

De maneira a suportar a realização das experiências acima explicadas, foram criados vários instrumentos de apoio à execução das mesmas.

3.2.2.1 Diagramas de actividades

Para que os participantes nas experiências, que estavam a ter a sua primeira abordagem aos DA, conseguissem percepcionar os principais conceitos de um DA, foi criado um DA exemplo.

O DA "Pedido de proposta de orçamento" representado na Figura 9 descreve um cenário onde um cliente visita uma empresa e solicita uma proposta de orçamento (anexo I). O pedido é feito ao funcionário comercial e se o valor previsto desse pedido estiver acima dos 500€ a proposta segue para o director comercial de maneira a ser este a prepará-la. Se o valor do pedido de proposta de orçamento estiver abaixo ou for igual a 500€ é o próprio funcionário comercial que prepara a proposta. Em ambos os casos é o funcionário comercial que envia a proposta de orçamento ao cliente.

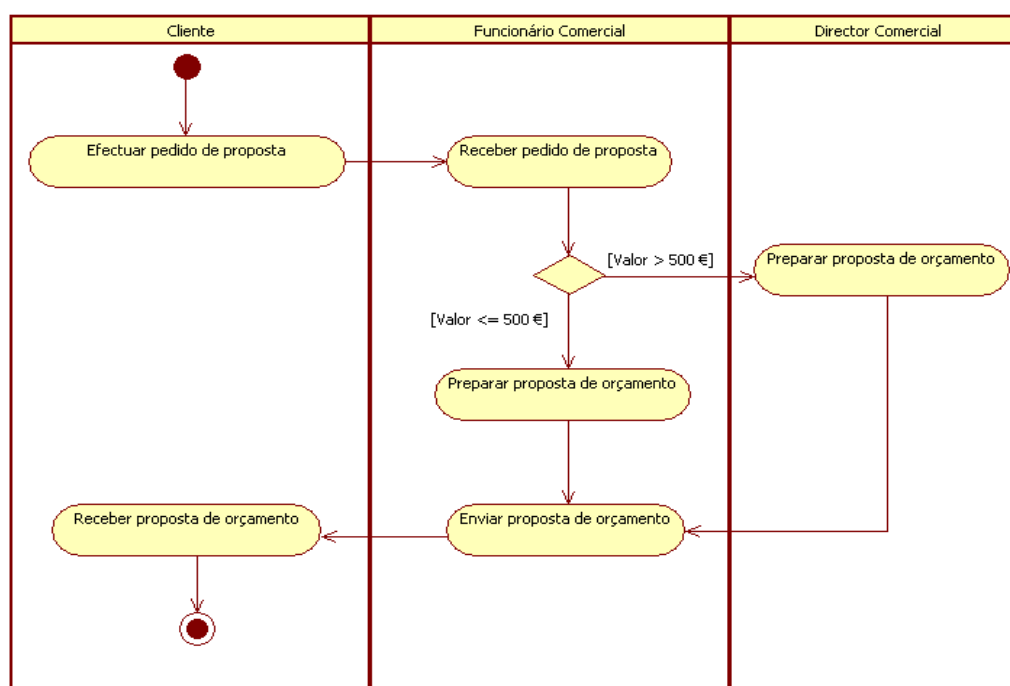



Figura 9 - Diagrama de Actividades "Pedido de proposta de orçamento"

O DA “Pedido de justificação de faltas” representado na Figura 10, foi criado com o propósito de apoiar os investigadores, em relação à análise/correcção do problema/exercício proposto realizado pelos participantes na experiência. O enunciado do problema/exercício proposto encontra-se no anexo II.

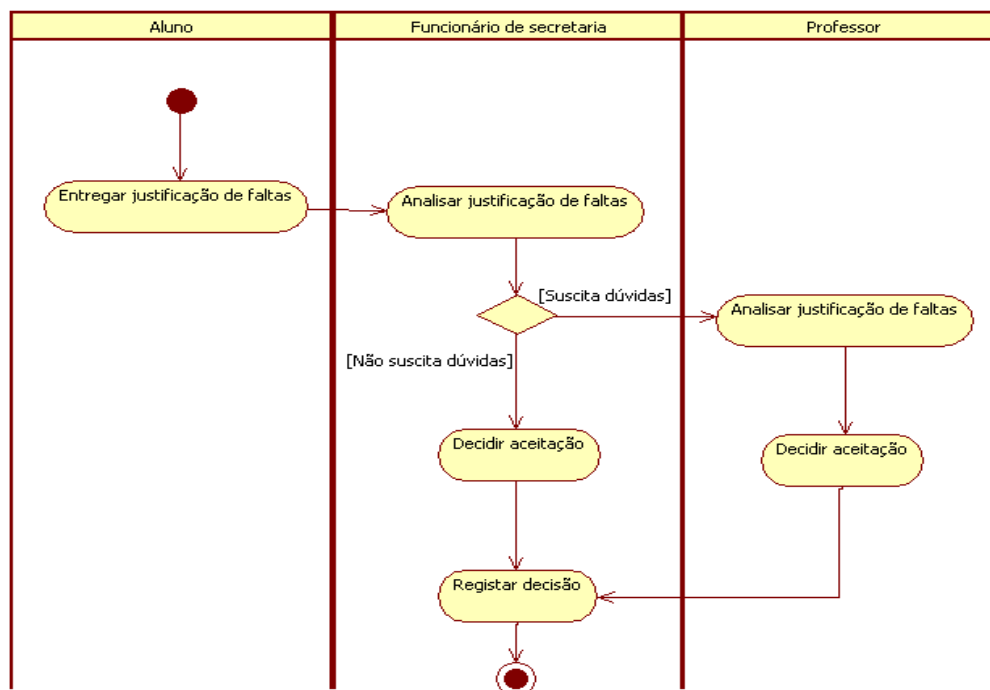



Figura 10 - Diagrama de Actividades “Pedido de justificação de faltas”

Este diagrama descreve um cenário no qual um estudante se desloca à secretaria de uma escola e entrega a sua justificação de faltas. A justificação é entregue ao funcionário da secretaria e é feita uma análise por parte deste. Se a justificação suscitar dúvidas é enviada para o professor da disciplina para análise, sendo que a decisão de aceitar é da sua responsabilidade. Se a justificação não suscitar dúvidas, a decisão pertence ao funcionário da secretaria que em qualquer um dos casos regista sempre a decisão.

3.2.2.2 Manual sintético

A ideia do uso de um manual que explicitasse os principais componentes dos DA surgiu com o desígnio de apoiar os participantes das experiências que necessitavam de resolver o problema proposto. Deste modo foi criado um manual sintético (anexo III) com imagens e uma breve explicação dos principais elementos dos DA.

3.2.2.3 Ferramenta de simulação

De forma a poderem avaliar-se os diferentes aspectos pertinentes, foram criados doze cenários diferentes, representados na Tabela 6, usando a ferramenta Microsoft PowerPoint. Como se pode observar na Figura 11, foram utilizadas as potencialidades da ferramenta para representar o que seria um exemplo do resultado pretendido após o desenvolvimento do sistema. Assim, foi possível apresentar na sequência desejada e em diferentes espaços de tempo, os actores, as actividades e os pontos de decisão, bem como os pontos inicial e final, dando desta forma aos alunos participantes nas experiências a possibilidade de observarem o processo de um pedido de orçamento de uma forma diferente daquela representada sobre a forma de DA.

Pretendia-se com isto mostrar as diferentes actividades espaçadas através de intervalos de tempo e ainda simular a passagem de informação através os diferentes intervenientes no processo simulado entre cada actividade concluída.

Com diferentes variantes deste diapositivo, foi possível criar os diferentes cenários, possibilitando assim, de acordo com cada aspecto em análise em determinada experiência, analisar a importância da existência de som, o *fade-out* total ou parcial das actividades e o aparecimento dos diferentes cenários[1] – após um ponto de decisão – na mesma sequência de actividades ou em sequências distintas.

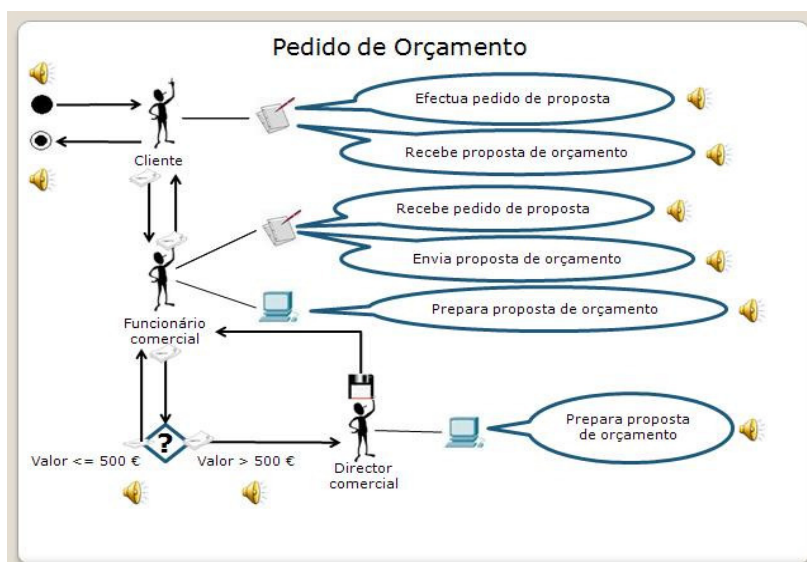



Figura 11 - Simulação do Diagrama de Actividades "Pedido de proposta de orçamento" usando a ferramenta Microsoft PowerPoint

[1] Para evitar equívocos em torno do termo *cenários,* lembramos que é possível fazer uso do mesmo para nos referirmos a diferentes percursos na progressão de um processo de negócio representado num DA, após a passagem por um ponto de decisão, ou aos vários cenários avaliados no total das experiências realizadas.

Tabela 6 - Cenários criados usando a ferramenta Microsoft PowerPoint

Cenários	Descrição
1	Pedido de proposta de orçamento com dois cenários, com som e sem *fade-out* de actividades.
2	Pedido de proposta de orçamento com dois cenários, sem som e sem *fade-out* de actividades.
3	Pedido de proposta de orçamento com dois cenários, com som e com *fade-out* de actividades.
4	Pedido de proposta de orçamento com dois cenários, sem som e com *fade-out* de actividades.
5	Pedido de proposta de orçamento com o primeiro cenário, com som e sem *fade-out* de actividades.
6	Pedido de proposta de orçamento com o primeiro cenário, com som e com *fade-out* de actividades.
7	Pedido de proposta de orçamento com o primeiro cenário, sem som e sem *fade-out* de actividades.
8	Pedido de proposta de orçamento com o primeiro cenário, sem som e com *fade-out* de actividades.
9	Pedido de proposta de orçamento com o segundo cenário, com som e sem *fade-out* de actividades.
10	Pedido de proposta de orçamento com o primeiro cenário, com som e com *fade-out* de actividades.
11	Pedido de proposta de orçamento com o primeiro cenário, sem som e sem *fade-out* de actividades.
12	Pedido de proposta de orçamento com o primeiro cenário, sem som e com *fade-out* de actividades.

O capítulo três teve como objectivo apresentar o processo que levou à eleição do método de investigação e técnicas a seguir e ainda aos instrumentos que foram pensados e criados para que o procedimento de realização de experiências, a respectiva análise dos dados recolhidos e o processo de construção da ferramenta fossem facilitados.

As conclusões que serão efectuadas no decorrer do próximo capítulo tiveram como base de sustentação este capítulo e são fruto do trabalho aqui descrito.

4 Sistema de apoio ao ensino de diagramas de actividades da UML – Decisões e cenários alternativos

O trabalho desenvolvido nos capítulos anteriores é de extrema importância para a compreensão do sistema apresentado. Assim, é no capítulo que agora se inicia que está o foco do tema tratado nesta dissertação.

Será descrito de seguida o processo de realização das experiências realizadas com vista ao levantamento dos requisitos necessários para o sistema, apresentados os dados obtidos, bem como as conclusões que a sua análise nos possibilita. São também apresentadas as tecnologias utilizadas e descrito o desenvolvimento do editor de diagramas e da ferramenta de animação dos mesmos.

Uma vez identificado o problema deste projecto, os objectivos e as motivações do mesmo, o método de investigação e as técnicas a aplicar, é necessário colocar em prática os resultados obtidos até aqui, de forma a podermos atingir os objectivos a que nos propusemos: desenvolver um sistema de apoio ao ensino de DA da UML. Para isso é necessário ter em conta as conclusões alcançadas, nomeadamente o papel da ES no desenvolvimento de SI, a simplificação do processo de concepção de software fornecida pela UML e as vantagens do uso dos DA.

Como se pretende desenvolver software de simulação de apoio ao ensino, é muito importante ter em consideração a sintaxe dos DA e os cuidados a ter no tratamento da informação que estes pretendem transmitir, especialmente no tratamento das decisões e

cenários alternativos garantindo que a informação é clara e objectiva, aproximando assim os alunos do ponto de entendimento pretendido.

4.1 Recolha de dados das experiências

Tendo em conta o método de investigação escolhido, método de Investigação-Acção e identificadas as vantagens da realização de experiências como técnica de investigação, torna-se fundamental criar condições para um correcto registo dos dados de cada experiência. É extremamente importante que os investigadores possam consultar, as vezes que acharem necessárias, os dados registados e é igualmente importante que quando estes o façam, possam como que recriar, um ambiente o mais aproximado possível do momento de realização das experiências.

De acordo com o método de investigação adoptado, as experiências foram planeadas e realizadas segundo a Figura 12.

Figura 12 - Modelo cíclico de Investigação-Acção

De uma forma simplista, o método Investigação-Acção consiste na identificação de um problema, no desenvolvimento de uma solução que resolva esse problema e se após essa acção os esforços não forem satisfatórios, gera-se um novo ciclo com um novo plano de acção, até que o problema inicial seja resolvido.

Assim, cada experiência define um ciclo ao longo de todo o processo de investigação, tendo cada uma delas, um plano de acção, a concretização da acção com uma observação sistemática da mesma, seguindo-se a fase de reflexão que determina a necessidade de gerar uma nova iteração.

Usufruindo do material de suporte à realização das experiências – DA "Pedido de proposta de orçamento" e "Pedido de justificação de faltas", manual sintético e ferramenta de simulação – estas foram estruturadas sequencialmente, de acordo com o método de investigação adoptado.

Pré-experiência (PE):

1. Disponibilizar ao aluno um diagrama de actividades exemplo;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Mostrar ao aluno a animação sem som, com cenários juntos e separados e com *fade-out* total e parcial das actividades;
4. Pedir ao aluno que explique o que percebeu daquilo que viu;
5. Mostrar ao aluno a animação com som, com cenários juntos e separados e com *fade-out* total e parcial das actividades;
6. Pedir ao aluno para explicar o que percebeu daquilo que viu;
7. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
8. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 1 (E1):

1. Disponibilizar ao aluno um diagrama de actividades exemplo;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Mostrar ao aluno a animação sem som, com cenários juntos e com *fade-out* parcial das actividades;
4. Pedir ao aluno que explique o que percebeu daquilo que viu;
5. Mostrar ao aluno a animação com som, com cenários juntos e com *fade-out* parcial das actividades;
6. Pedir ao aluno para explicar o que percebeu daquilo que viu;
7. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
8. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 2 (E2):

1. Disponibilizar ao aluno um diagrama de actividades exemplo;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Mostrar ao aluno a animação com som, com cenários juntos e com *fade-out* parcial das actividades;
4. Pedir ao aluno que explique o que percebeu daquilo que viu;
5. Mostrar ao aluno a animação com som, com cenários separados e com *fade-out* parcial das actividades;
6. Pedir ao aluno para explicar o que percebeu daquilo que viu;
7. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
8. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 3 (E3):

1. Disponibilizar ao aluno um diagrama de actividades exemplo;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Mostrar ao aluno a animação com som, com cenários separados e com *fade-out* parcial das actividades;
4. Pedir ao aluno que explique o que percebeu daquilo que viu;
5. Mostrar ao aluno a animação com som, com cenários separados e com *fade-out* total das actividades;
6. Pedir ao aluno para explicar o que percebeu daquilo que viu;
7. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
8. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 4 (E4):

1. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
2. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 5 (E5):

1. Disponibilizar ao aluno um diagrama de actividades exemplo;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;

4. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 6 (E6):

1. Mostrar ao aluno a animação com som, com cenários separados e com *fade-out* parcial das actividades;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
4. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 7 (E7):

1. Disponibilizar ao aluno um diagrama de actividades exemplo;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Mostrar ao aluno a animação com som, com cenários separados e com *fade-out* parcial das actividades;
4. Pedir ao aluno para explicar o que percebeu daquilo que viu;
5. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
6. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

Experiência 8 (E8):

1. Mostrar ao aluno a animação com som, com cenários separados e com *fade-out* parcial das actividades;
2. Pedir ao aluno para explicar o que percebeu daquilo que viu;
3. Disponibilizar ao aluno um diagrama de actividades exemplo;
4. Pedir ao aluno para explicar o que percebeu daquilo que viu;
5. Disponibilizar ao aluno um manual sintético sobre diagramas de actividades;
6. Disponibilizar ao aluno um exercício para resolução sobre diagramas de actividades.

No decorrer de cada experiência e sempre que era solicitado ao aluno que explicasse o seu entendimento do que ia observando, as suas respostas eram registadas; sempre que a experiência continha a realização de um exercício para resolução, o resultado do mesmo era avaliado e cotado, registando também o tempo da sua realização. Os dados registados em todas as experiências são apresentados na Tabela 7 e Tabela 8.

Cada experiência foi registada em vídeo, som e papel, incluindo o registo em papel dos exercícios elaborados pelos alunos, para posterior análise e desta forma ser possível avaliar a evolução da compreensão dos DA apresentados ao longo das experiências.

Tabela 7 - Respostas e avaliação dos resultados das experiências (Primeira fase)

Temas em análise	PE-A	PE-B	E1-A	E1-B	E1-C	E1-D	E2-A	E2-B	E2-C	E2-D	E3-A	E3-B	E3-C	E3-D
Compreensão do diagrama apresentado	4	2	5	5	3	4	4	5	3	4	5	5	3	3
As animações permitiram compreender alguns aspectos que não estavam claros à partida	*	*	N	N	S	N	S	N	S	S	N	N	S	S
É preferível visualizar as animações com som relativamente a animações sem som	*	*	S	S	S	S	*	*	*	*	*	*	*	*
É preferível visualizar as animações com cenários separados relativamente a cenários juntos	*	*	*	*	*	*	N	S	S	N	N	S	S	N
É preferível visualizar as animações com *fade-out* total relativamente a *fade-out* parcial	*	*	*	*	*	*	*	*	*	*	N	I	S	I
Depois de ver todas as representações, o caso foi bem explicitado	5	5	5	5	4	5	5	5	4	5	4	5	5	5
Criação do diagrama (Pedido de justificação de faltas)	2	3	5	3	2	3	2	5	3	5	4	3	2	2
Tempo usado para a criação de um diagrama (Pedido de justificação de faltas) (mm:ss)	07:00	03:00	06:00	08:02	07:38	08:12	06:52	10:50	11:12	08:01	15:11	07:13	06:53	10:08
Legenda:	PE – Pré-Experiência E – Experiência						S – Sim N – Não I – Indiferente				1 – Muito Mau 2 – Mau 3 – Razoável 4 – Bom 5 – Excelente			

Tabela 8 - Respostas e avaliação dos resultados das experiências (Segunda fase)

Temas em análise	E4-A	E4-B	E4-C	E5-A	E5-B	E5-C	E6-A	E6-B	E6-C	E7-A	E7-B	E7-C	E8-A	E8-B	E8-C
Compreensão do diagrama apresentado	*	*	*	5	5	4	*	*	*	3	1	2	2	5	5
As animações permitiram compreender alguns aspectos que não estavam claros à partida	*	*	*	*	*	*	*	*	*	S	N	S	*	*	*
É preferível visualizar as animações com som relativamente a animações sem som	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
É preferível visualizar as animações com cenários separados relativamente a cenários juntos	*	*	*	*	*	*	N	S	N	N	N	N	N	N	N
É preferível visualizar as animações com *fade-out* total relativamente a *fade-out* parcial	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Depois de ver todas as representações, o caso foi bem explicitado	*	*	*	*	*	*	3	5	5	5	2	5	2	5	5
Criação do diagrama (Pedido de justificação de faltas)	2	1	3	2	1	3	1	3	4	3	4	5	4	3	2
Tempo usado para a criação de um diagrama (Pedido de justificação de faltas) (mm:ss)	09:15	07:13	08:18	09:54	06:08	10:10	04:45	02:14	06:42	05:39	07:55	06:50	09:53	04:35	09:25
Legenda:	PE – Pré-Experiência E – Experiência						S – Sim N – Não I – Indiferente				1 – Muito Mau 2 – Mau 3 – Razoável 4 – Bom 5 – Excelente				

4.2 Análise de dados recolhidos

A recolha de dados das experiências foi bastante pormenorizada, com intuito de se obter uma análise conclusiva no final de cada experiência. Foi feita a análise dos resultados para reter de imediato as conclusões possíveis e assim avançar com o grupo de experiências[1] seguinte, correspondendo este a um ciclo iterativo do método de Investigação-Acção.

Após a realização das pré-experiências foi possível identificar de que forma seriam estruturadas as experiências seguintes e qual seria o foco da investigação de cada uma delas. Uma vez terminadas as experiências E1[2] foi possível verificar, como podemos observar na Figura 13, que a totalidade dos alunos prefere que as animações contenham som.

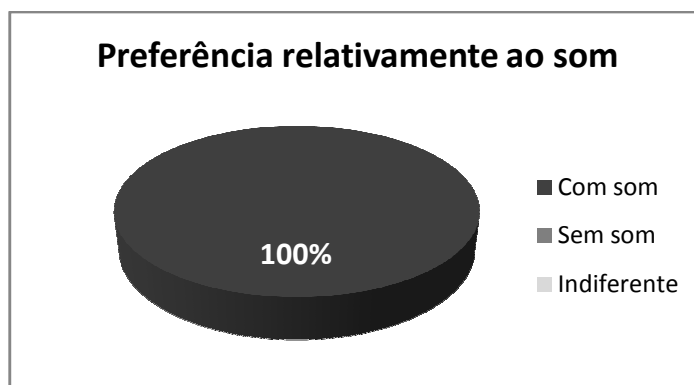



Figura 13 - Percentagens de preferências relativamente à presença de som

Depois de analisados os dados das experiências E2, pudemos constatar que a grande maioria dos alunos inquiridos prefere a apresentação conjunta dos diferentes cenários do diagrama, ou seja, como demonstra a Figura 14, a permanência dos vários cenários de um diagrama na sua animação facilita a compreensão do processo.

---

[1] Entenda-se por grupo de experiências, aquelas que obedecem à mesma estrutura e cuja realização é repetida com alunos diferentes.

[2] E1 é o código adoptado para nos referirmos às experiências do grupo 1, que, neste caso, pretendiam avaliar as vantagens/desvantagens da utilização do som na animação dos diagramas desenvolvidos pelos utilizadores. Estes códigos podem ser observados nas tabelas 7 e 8, sendo aí acrescentada uma letra ao código, para identificar uma única experiência no grupo E1.

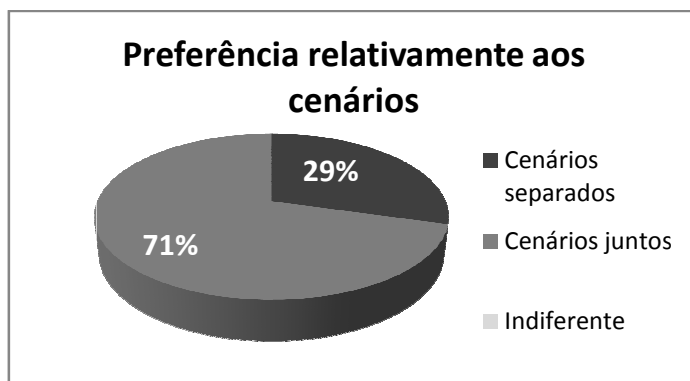



Figura 14 - Percentagens de preferências relativamente à apresentação dos diferentes cenários

Em relação às experiências E3, pudemos verificar a distribuição das opiniões dos alunos relativamente ao *fade-out* total ou parcial das actividades já decorridas na animação, isto é, conforme a Figura 15, para ¼ dos alunos é preferível eliminar totalmente as actividades, o mesmo número de alunos preferem que as actividades se mantenham na animação após a sua amostragem e 50% dos alunos referiram ser-lhes indiferente. A Figura 16 e a Figura 17 demonstram o aspecto da simulação apresentada aos alunos com *fade-out* total e *fade-out* parcial respectivamente.

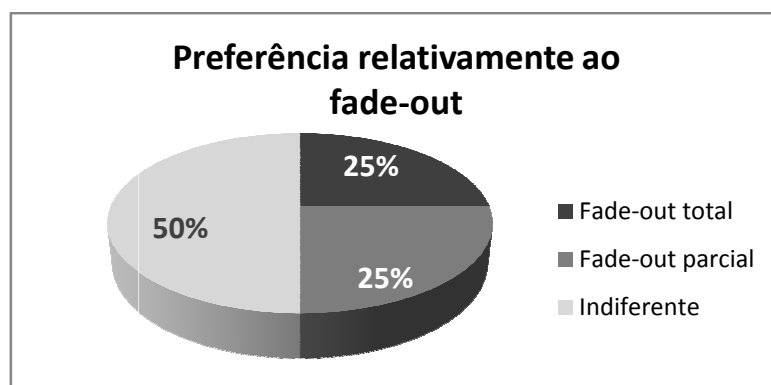



Figura 15 - Percentagens de preferências dos alunos relativamente ao *fade-out* das actividades

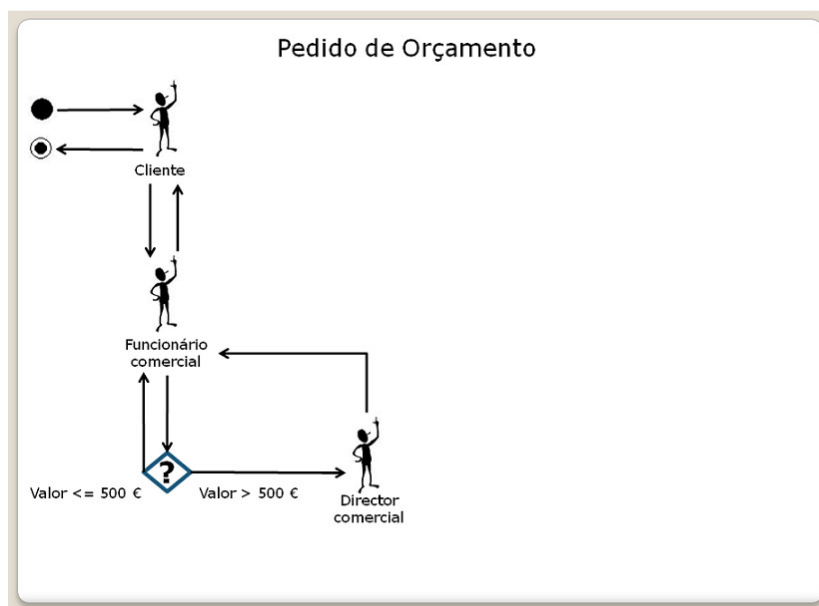



Figura 16 - Aspecto final da simulação com *fade-out* total das actividades

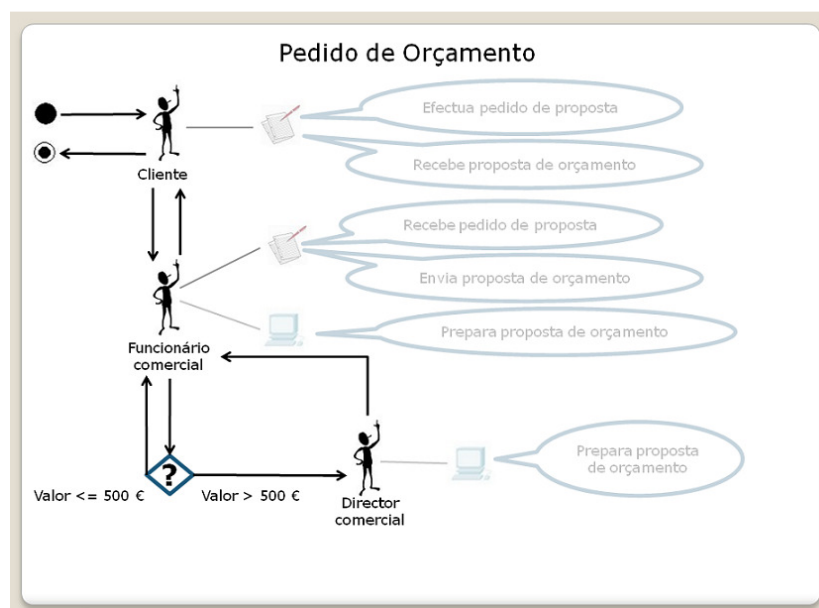



Figura 17 - Aspecto final da simulação com *fade-out* parcial das actividades

Contudo, para que no final de cada animação, seja possível consultar toda a sequência de actividades realizadas no decorrer do processo, optamos por manter todas as actividades na animação do diagrama criado pelo utilizador do sistema e assim facilitar o entendimento geral do mesmo.

Quanto às experiências E4, podemos observar na Tabela 8, que sem qualquer visualização de DA exemplo, ou qualquer animação representativa do mesmo, os alunos tiveram mais dificuldade em elaborar o exercício final, que consistia em representar sobre a forma de DA o processo de ”Pedido de justificação de faltas”, que lhes foi fornecido em texto, tendo-se obtido 2 valores numa escala de 1 a 5 para a média de resultados.

Na Tabela 8, também podemos verificar, relativamente às experiências E5, onde apenas foi mostrado aos alunos um DA exemplo e não houve qualquer amostragem de animações, que a média de resultados do exercício final foi 2 valores numa escala de 1 a 5, enquanto nas experiências E6, onde só foi possível aos alunos visualizarem uma animação de um DA exemplo, os resultados do exercício final atingiram a média de 2,67 valores na mesma escala.

Ainda na Tabela 8, é possível reter informação que nos leva a concluir que, no decorrer da experiência onde é mostrada ao aluno uma animação de um DA, o resultado do exercício final é sempre superior ao resultado do mesmo exercício, quando tal animação não é observada. Nas experiências E7 e E8 é importante realçar os resultados dos exercícios finais, 4 e 3 valores respectivamente, independentemente da ordem pela qual são observados o diagrama e a animação.

Na Figura 18 está reflectida a opinião dos alunos inquiridos, relativamente à contribuição da visualização de animações na compreensão do processo. Podemos observar que a maioria dos alunos refere que as animações permitem compreender aspectos que não estavam claros antes da observação das mesmas.

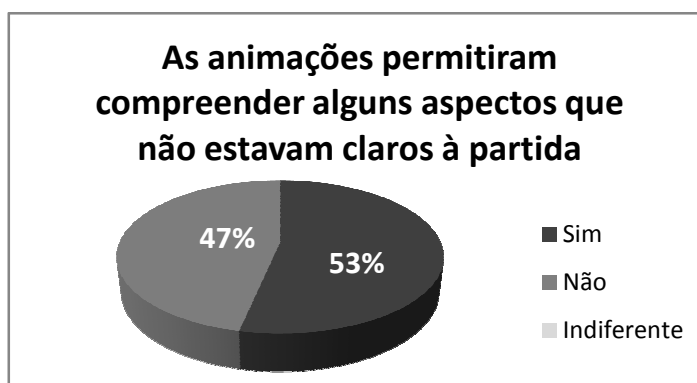



Figura 18 - Percentagens relativas à contribuição das animações na percepção do processo

Com base nestes dados e nas conclusões tiradas a partir deles, foi desenvolvido o sistema com vista a auxiliar o ensino de DA. Para isso foi feita a modelação do sistema e posteriormente o desenvolvimento da ferramenta.

4.3 Concepção da solução

A secção que agora se inicia esclarece o processo de modelação e construção da aplicação desenvolvida. Desta maneira, nesta secção estão especificados os requisitos funcionais e não funcionais da solução, são especificados sob a forma de diagramas de casos-de-uso e descritos em texto e é explicada a forma de concepção da aplicação final bem como as tecnologias usadas.

4.3.1 Modelação do sistema

A primeira medida de sucesso de qualquer sistema de software é tirada a partir do grau de desempenho que corresponde ao propósito para o qual foi intencionado (Nuseibeh, et al., 2000). Assim, o levantamento de requisitos de um sistema deve ser a primeira medida a ser tomada quando se deseja desenvolver algo, pois este é o processo de descobrir o propósito do software através da identificação das partes interessadas no mesmo e nas suas necessidades, documentando-as para possibilidade de análise, comunicação e posterior execução (Nuseibeh, et al., 2000).

A especificação de requisitos é, na maior parte dos casos, o primeiro passo da construção de qualquer software (Goguen, et al., 1994).

Pareceu-nos também importante concretizar uma especificação de requisitos funcionais de modo a que nos guiasse na construção da aplicação e tivéssemos sempre presente os objectivos a concretizar.

Deste modo, foi elaborada a seguinte lista de requisitos funcionais, a qual nos diz que a aplicação deverá:

- Permitir ao utilizador criar um diagrama;
- Permitir ao utilizador guardar um diagrama;
- Permitir ao utilizador pré-visualizar um diagrama;
- Permitir ao utilizador imprimir um diagrama;
- Permitir ao utilizador visualizar ajuda;

- Permitir ao utilizador alterar o zoom de um diagrama;
- Permitir ao utilizador visualizar código-fonte do diagrama;
- Permitir ao utilizador visualizar opções de simulação;
- Permitir ao utilizador alterar opções de simulação;
- Permitir ao utilizador visualizar simulação.

De modo a complementar a especificação de requisitos foi também criada uma lista de requisitos não-funcionais, ou seja, restrições no sistema a ser desenvolvido e restrições no processo de desenvolvimento (Kotonya, et al., 1996). Assim, a aplicação deverá:

- Ser implementada na linguagem HTML;
- Usar a linguagem de programação JavaScript para o desenho de diagramas e posterior simulação;
- Usar a linguagem PHP e o servidor Apache como *backend*,
- Usar a linguagem XML para armazenamento dos dados do diagrama e configuração do *backend*.

De modo a consolidar toda a especificação do sistema, afigurou-se útil a criação dos diagramas de casos-de-uso do sistema.

Os casos-de-uso têm como objectivo principal a captura das funcionalidades do sistema tal como é visto pelos utilizadores e é geralmente construído nos primeiros estágios de desenvolvimento de qualquer sistema de software (Rumbaugh, et al., 1999).

Rumbaugh (Rumbaugh, et al., 1999) refere ainda outros objectivos dos casos de uso, assim um diagrama de casos de uso é geralmente criado para:

- Guiar a concepção do sistema (captura de requisitos funcionais);
- Guiar a implementação do sistema (implementação de um sistema que satisfaça os requisitos);
- Guiar o processo de testes (testar os requisitos que são satisfeitos).

Desta forma foi criado o diagrama de casos-de-uso representado na Figura 19.

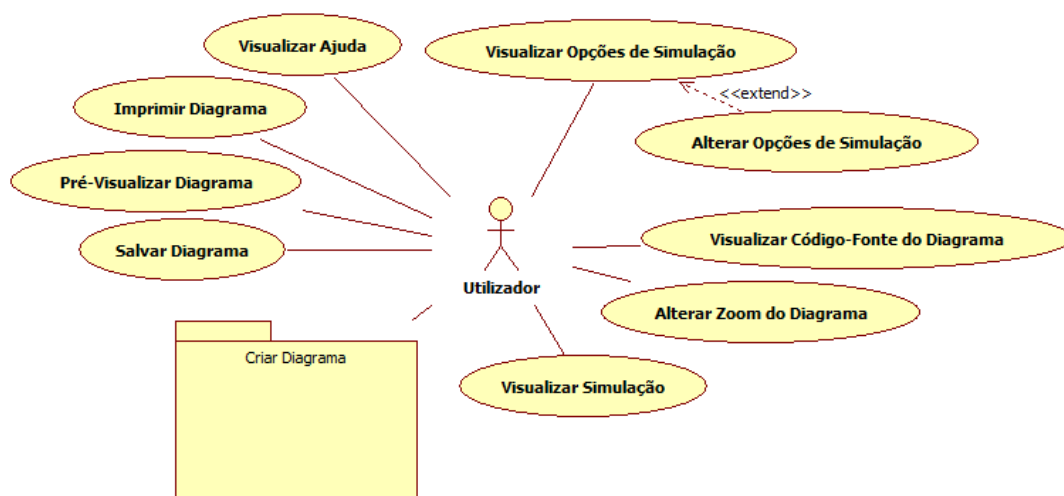



Figura 19 - Diagrama de casos-de-uso do sistema

O diagrama principal, apresentado na Figura 19 apresenta o *package* "Criar Diagrama" que concentra em si todas as funcionalidades respeitantes ao próprio acto de criação do DA. A Figura 20 apresenta o detalhe desse *package*.

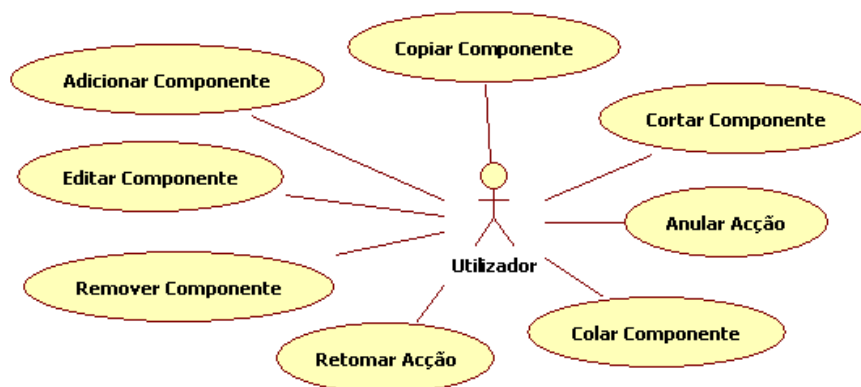



Figura 20 - Package Criar Diagrama

De forma a contemplar uma total percepção dos diagramas de casos-de-uso criados, foi também feita a sua descrição, estruturada com uma breve descrição do caso-de-uso, as pré-condições para que o caso-de-uso se realize, o fluxo primário, o fluxo alternativo, as excepções e as pós-condições.

Caso-de-uso “Salvar Diagrama”	
ID	UC1
Breve Descrição	Permite ao utilizador salvar um diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona a opção “Salvar” da *toolbar*. 2. Um ficheiro XML é criado no *backend*.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O diagrama é salvo e um ficheiro XML é criado no *backend*.

Caso-de-uso “Pré-Visualizar Diagrama”	
ID	UC2
Breve Descrição	Permite ao utilizador pré-visualizar um diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona a opção “Pré-visualizar” da *toolbar*. 2. A aplicação mostra o diagrama que está a ser criado num novo separador/página.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O utilizador visualiza um diagrama num novo separador/página.

Caso-de-uso “Imprimir Diagrama”	
ID	UC3
Breve Descrição	Permite ao utilizador imprimir um diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona a opção “Imprimir” da *toolbar*. 2. É mostrada uma caixa com opções de impressão.
Fluxo Alternativo	1. Se o utilizador seleccionar a opção “Cancelar”; 2. A caixa com opções de impressão é fechada.
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O diagrama é impresso.

Caso-de-uso “Visualizar Ajuda”	
ID	UC4
Breve Descrição	Permite ao utilizador visualizar tópicos de ajuda.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona a opção “Visualizar Ajuda” da *toolbar*. 2. A aplicação mostra tópicos de ajuda.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O utilizador visualiza tópicos de ajuda.

Caso-de-uso "Alterar Zoom do Diagrama"	
ID	UC5
Breve Descrição	Permite ao utilizador alterar o zoom do diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona uma das opções de zoom listadas. 2. A aplicação altera o zoom do diagrama que está a ser criado.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O utilizador visualiza o diagrama com diferente zoom.

Caso-de-uso "Visualizar Código-Fonte do Diagrama"	
ID	UC6
Breve Descrição	Permite ao utilizador visualizar o código-fonte do diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona a *check-box* "Visualizar código-fonte". 2. A aplicação mostra o código-fonte do diagrama que está a ser criado.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O utilizador visualiza o código-fonte do diagrama que está a criar.

Caso-de-uso "Visualizar Opções de Simulação"	
ID	UC7
Breve Descrição	Permite ao utilizador visualizar opções de simulação.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador visualiza todas as opções existentes para a simulação. 2. O utilizador escolhe as opções que pretende para a simulação.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	A aplicação volta à criação do diagrama.

Caso-de-uso "Visualizar Simulação"	
ID	UC8
Breve Descrição	Permite ao utilizador visualizar a simulação do diagrama criado.
Pré-Condições	O utilizador encontra-se no URL correcto e tem um diagrama criado.
Fluxo Primário	1. O utilizador clica no botão "Simular". 2. A aplicação mostra uma simulação do diagrama criado.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O utilizador visualiza uma simulação do diagrama criado.

Caso-de-uso "Adicionar Componente"	
ID	UC9
Breve Descrição	Permite ao utilizador adicionar uma componente ao diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona uma das componentes presentes na barra lateral e arrasta-a para a área de desenho. 2. A componente é acrescentada ao diagrama.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	

Caso-de-uso "Copiar Componente"	
ID	UC10
Breve Descrição	Permite ao utilizador copiar uma componente do diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona uma das componentes presentes na área de desenho copia-a utilizando o ícone da barra lateral.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	

Caso-de-uso "Cortar Componente"	
ID	UC11
Breve Descrição	Permite ao utilizador cortar uma componente do diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto.
Fluxo Primário	1. O utilizador selecciona uma das componentes presentes na área de desenho e corta-a utilizando o ícone da barra lateral. 2. A componente é eliminada do diagrama.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	O diagrama fica sem a componente cortada e as suas ligações desaparecem.

Caso-de-uso "Anular Acção"	
ID	UC12
Breve Descrição	Permite ao utilizador anular a última acção efectuada no diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto. O utilizador já tinha introduzido alterações na área de desenho.
Fluxo Primário	1. O utilizador selecciona o ícone de anular acção presente na barra lateral e a última acção é anulada. 2. A componente é eliminada do diagrama.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	

Caso-de-uso "Colar Componente"	
ID	UC13
Breve Descrição	Permite ao utilizador colar uma componente no diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto. O utilizador cortou/copiou uma componente já existente no diagrama.
Fluxo Primário	1. O utilizador encontra-se na área de desenho e cola uma componente utilizando o ícone da barra lateral. 2. A componente é acrescentada ao diagrama.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	

Caso-de-uso "Retomar acção"	
ID	UC14
Breve Descrição	Permite ao utilizador retomar a última acção anulada no diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto. O utilizador já tinha introduzido alterações na área de desenho.
Fluxo Primário	1. O utilizador selecciona o ícone de retomar acção presente na barra lateral e a última acção anulada é retomada.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	

Caso-de-uso "Editar Componente"	
ID	UC15
Breve Descrição	Permite ao utilizador editar uma componente presente no diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto. O utilizador já tinha introduzido alterações na área de desenho.
Fluxo Primário	1. Utilizando o botão direito do rato, o utilizador selecciona a opção de editar estilo da componente. 2. O utilizador insere o estilo pretendido para a componente seleccionada.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	

Caso-de-uso "Remover Componente"	
ID	UC16
Breve Descrição	Permite ao utilizador remover uma componente presente no diagrama.
Pré-Condições	O utilizador encontra-se no URL correcto. O utilizador já tinha introduzido alterações na área de desenho.
Fluxo Primário	1. O utilizador selecciona uma das componentes presentes na área de desenho e remove-a utilizando no ícone da barra lateral. 2. A componente é removida do diagrama.
Fluxo Alternativo	
Excepções	E1 – Em qualquer altura o utilizador muda de página/encerra o browser e sai da aplicação.
Pós-Condições	

A descrição dos casos-de-uso é de extrema importância, já que permite manter a coerência entre as funcionalidades que foram modeladas e as que são desenvolvidas, permitindo para isso esclarecer quaisquer dúvidas que possam surgir no momento do desenvolvimento de determinada funcionalidade.

4.3.2 Desenvolvimento do editor de diagramas

Como ferramenta de implementação do editor de DA, foi usado o mxGraph, uma biblioteca de funções em JavaScript que usa as capacidades dos browsers Web para disponibilizar o desenho interactivo de gráficos. Esta ferramenta foi escolhida pois supera as outras soluções existentes no tempo de arranque, interactividade e funcionalidades.

O mxGraph não necessita de instalação de *plug-ins* e a sua utilização pode ser feita a partir de uma ferramenta desenvolvida em Java, .Net, Hypertext PreProcessor (PHP) ou mesmo HyperText Markup Language (HTML). A interface do utilizador é feita em HTML e os dados são guardados e transferidos em eXtensible Markup Language (XML). Finalmente, o *backend* pode ser implementado usando Java, .NET ou PHP.

Neste caso, usando o mxGraph foi criado um editor em HTML, Figura 21, e algumas funções em JavaScript para efeitos de zoom, bem como edição de funções para editar as propriedades do diagrama e seus componentes. Foi também criado um ficheiro XML com as configurações da barra lateral, barra esta que disponibiliza as várias componentes que o utilizador dispõe para criar o diagrama e poder transferir dados com o servidor de maneira a visualizar a simulação. Podem ser consultadas outras figuras descritivas do editor de DA no anexo IV.

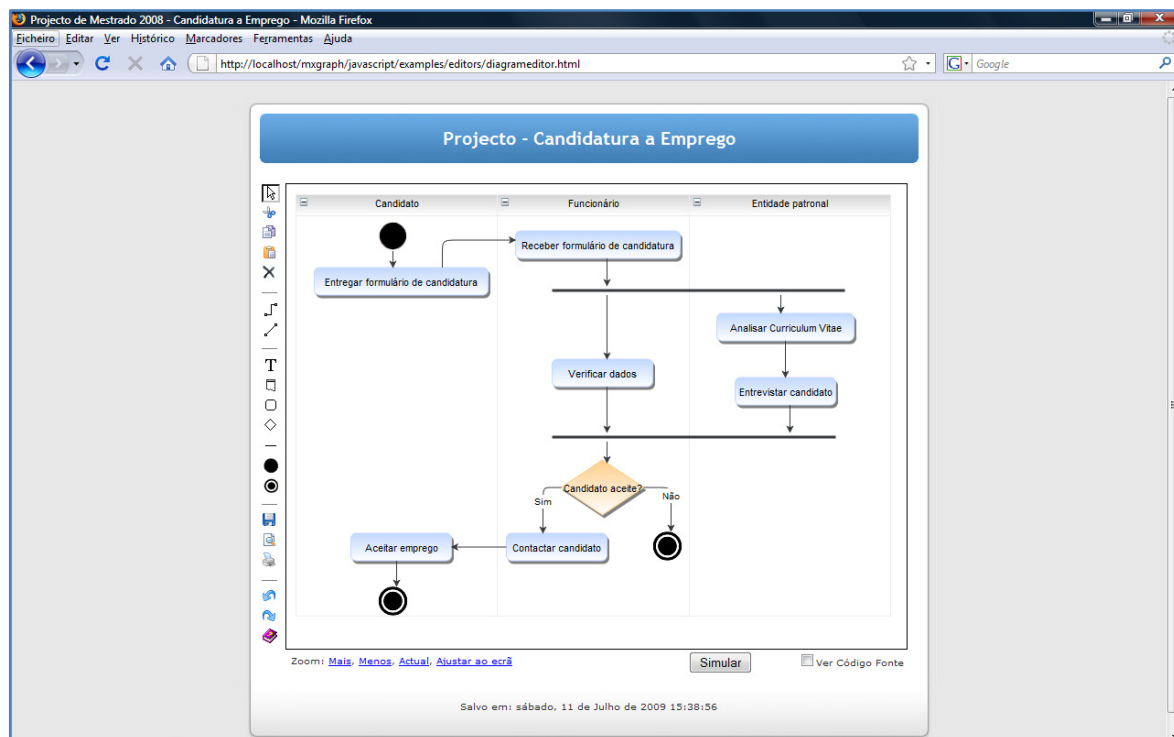



Figura 21 - Editor de Diagramas de Actividades

O *backend* deste editor de diagramas foi editado usando PHP, o que permite que seja criado um ficheiro temporário com os atributos que vão sendo adicionados ao diagrama à medida que este está a ser criado pelo utilizador.

4.3.3 Desenvolvimento da ferramenta de simulação

O objectivo central deste sistema é facilitar a compreensão e interpretação dos DA. Assim, é muito importante que o utilizador, ao visualizar a animação gerada a partir do DA que criou, identifique com rapidez e exactidão, as possíveis incorrecções do DA, permitindo-lhe validar e optimizar o processo de negócio que pretende representar no mesmo. Com permanente foco neste objectivo e com base na análise dos resultados das experiências realizadas anteriormente, foram tomadas algumas decisões. Assim, ao longo do desenvolvimento da ferramenta, foi tido em conta que:

- Uma animação deve iniciar-se sempre com a indicação de início acompanhada pelo respectivo símbolo da UML;
- Cada actor do sistema (*swimlane* ou barra de sincronização) deve ser representado na animação por uma de duas imagens distintas, dependendo do seu estereótipo (actor humano ou actor automático) e pela sua descrição;
- Cada actividade deve ser mostrada na animação, acompanhada pelo actor que a desempenha e pelo texto que a descreve;
- Cada actividade deve ser representada na animação por uma de duas imagens distintas, dependendo do seu estereótipo (actividade manual ou actividade digital);
- Cada conjunto (actor e actividade) deve ser apresentado em concordância com os respectivos símbolos da UML e respectiva descrição;
- Todas as actividades concorrentes, incluindo as que são desempenhadas por actores distintos, devem ser apresentadas em conjunto, num módulo destacado por cor diferente, cada uma alinhada com o respectivo actor;
- Os nós de decisão de uma animação devem ser apresentados por intermédio de um *pop-up* solicitando ao utilizador a escolha do cenário a ser seguido na animação;
- Uma animação deve terminar sempre com a indicação de fim acompanhada pelo respectivo símbolo da UML;
- A animação tem de manter obrigatoriamente a sequência de actividades que foi implementada no DA que lhe deu origem.

O desenvolvimento da ferramenta de simulação partiu da análise dos dados gerados pelo editor de DA. Esses dados são guardados num ficheiro XML aquando da criação de um DA, portanto, o primeiro passo foi fazer a análise cuidada do mesmo para se identificar a sua estrutura, elementos, atributos e tipo de dados dos atributos.

Feita a análise do ficheiro no qual são armazenados todos os dados necessários à construção da animação do DA, tornou-se possível fazer o *parsing* dos mesmos e assim ser possível identificar o ponto inicial e final da animação, bem como a sequência dos componentes que constituem o DA e o estereótipo de cada um desses componentes, já que diferentes tipos de componentes são mostrados de diferentes formas e recorrendo a diferentes técnicas e imagens.

Numa primeira versão, a linguagem de programação utilizada para desenvolvimento da ferramenta de simulação foi PHP (Figura 22), por se tratar de uma linguagem livre e familiar. Contudo acabamos por abandonar esta versão em detrimento de uma segunda, desenvolvida

em *JavaScript*. Esta decisão deveu-se essencialmente ao facto de a linguagem *JavaScript*, para além de ser ideal para dinamizar uma página HTML estática, tem uma enorme versatilidade para lidar com ambientes em árvore, como é o caso do ficheiro de dados criado em XML.

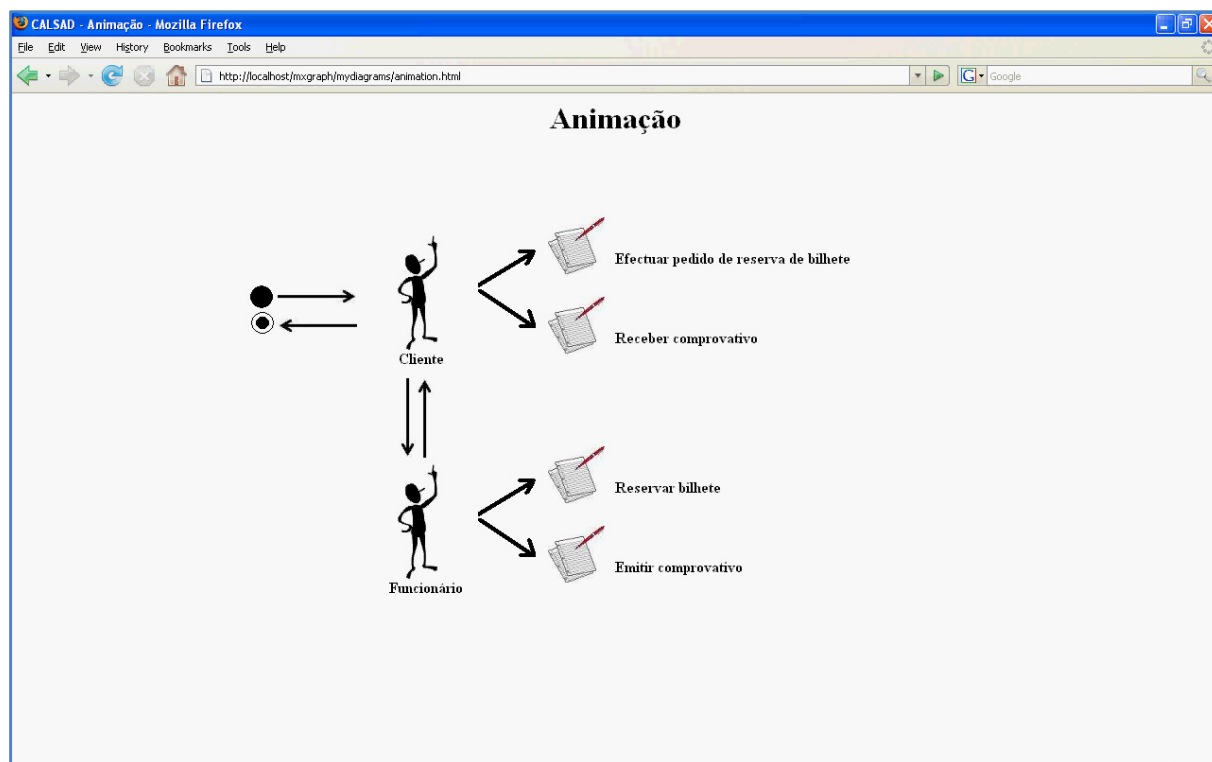



Figura 22 - Animação de um Diagrama de Actividades exemplo (versão 1 - PHP)

Desta forma, utilizando o método *ActiveXObject()* para criar um objecto do tipo *XMLDOM*, foi feito o *parsing* dos dados do ficheiro XML e com a versatilidade de métodos disponibilizados, tornou-se simples programar a ferramenta para aceder a qualquer elemento do XML e assim manter a sequência de amostragem dos componentes do diagrama.

Na segunda versão da aplicação, as preocupações centraram-se na compreensão, por parte do utilizador, das animações geradas e na importância de permitir-lhe fazer uma correcta interpretação das mesmas. Por este facto, o esquema de construção das animações foi alterado (Figura 23), abstraindo assim o utilizador do formato, esquematização e localização dos componentes num DA e permitindo-lhe seguir sem qualquer possibilidade de erro, a sequência traduzida pelo DA criado anteriormente, criando também para esse efeito pausas no tempo após a amostragem de cada actividade.

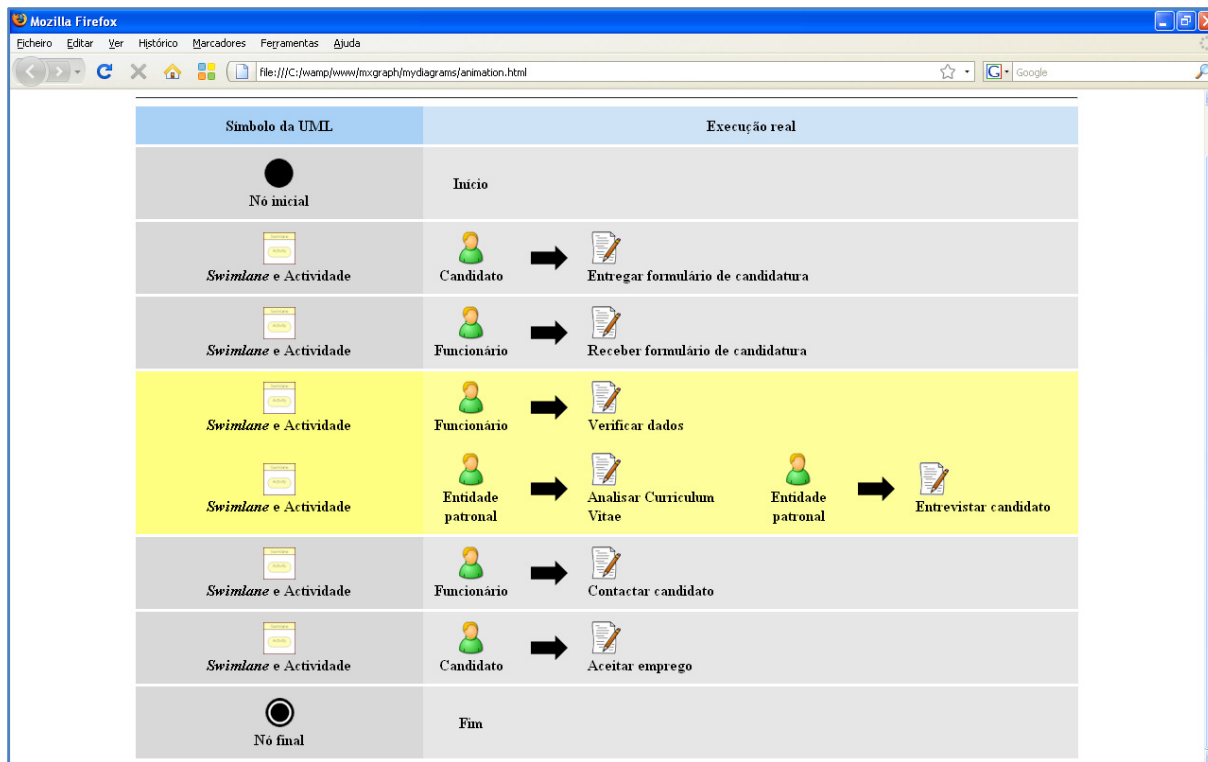



Figura 23 - Animação de um Diagrama de Actividades exemplo (versão 2 - JavaScript)

Com a implementação desta versão da ferramenta chegamos à conclusão de que neste caso não faria sentido manter todos os cenários, após passagem por um nó de decisão, na mesma animação. Esta decisão deve-se ao facto de a percentagem de ocupação do ecrã ser elevada, o que impossibilita a amostragem paralela dos diferentes cenários e ao facto de a sequência de amostragem só terminar com a chegada a um nó final, o que mentaliza o utilizador da obtenção do estado final da execução das actividades, tornando-se assim inesperado o reinício do novo cenário a partir do nó de decisão anterior. Podem ser consultadas outras figuras descritivas do processo de simulação de um DA no anexo V.

4.3.3.1 Funcionalidade de suporte aos nós de decisões na ferramenta de simulação

Relativamente à animação dos nós de decisão, após a opção de incluir apenas um cenário, como se descreve anteriormente, é necessário que a ferramenta seja programada para possibilitar a visualização de um cenário de cada vez que esta é executada. No entanto como é necessário que todos os cenários sejam visualizados, essa escolha caberá ao utilizador, apresentando-lhe uma mensagem no ecrã onde este pode ler a interrogação de entrada no nó de decisão e as várias condições de guarda de saída do nó de decisão (Figura 24).

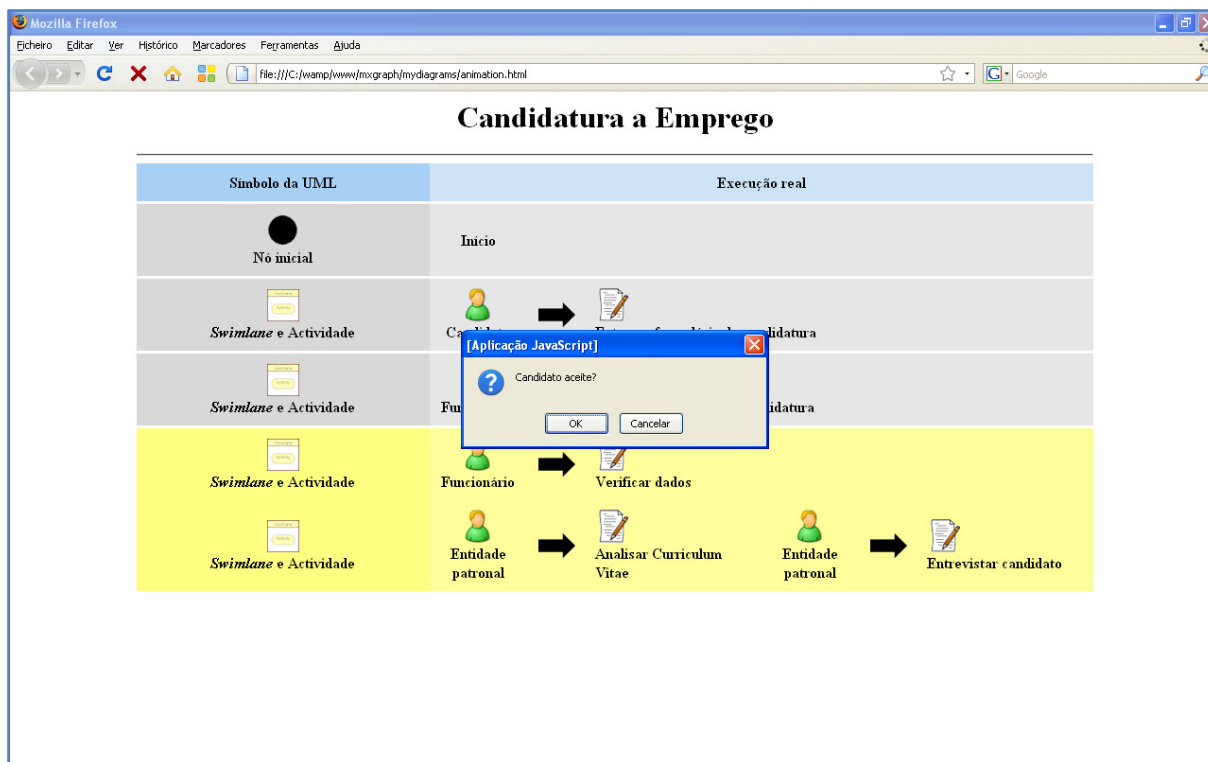



Figura 24 - Detalhe da animação de nós de decisão

Assim basta que o utilizador seleccione uma opção das apresentadas e automaticamente é visualizado o cenário que corresponde à condição de guarda seleccionada.

Concluindo o capítulo de abordagem ao desenvolvimento da ferramenta de simulação, importa lembrar que a mesma foi desenvolvida com intuito de auxiliar o ensino de DA da UML e é nesse sentido que aponta esta dissertação. Assim, e sempre numa perspectiva de desenvolvimento subsequente que melhore a robustez da ferramenta, podemos dizer que a aplicação apresentada respeita a maioria dos requisitos identificados inicialmente. Foram apresentadas algumas alterações, devidamente justificadas, nomeadamente a apresentação dos cenários alternativos e o *fade-out* das actividades, no entanto, reconhecemos na ferramenta uma base de lançamento para a investigação e exploração das TI a aplicar no ensino, deste e de outros diagramas da UML.

5 Considerações finais

No sentido de complementar o trabalho realizado ao longo deste estudo é importante concluir esta dissertação com algumas considerações acerca dos temas tratados em cada capítulo, discutir os resultados atingidos, apresentar as dificuldades encontradas, propor melhorias e novas funcionalidades a implementar na ferramenta de simulação e abordar os contributos de um sistema de apoio ao ensino de DA da UML.

5.1 Síntese da dissertação

Actualmente as TI são parte integrante da sociedade, principalmente ao nível empresarial e organizacional, dependendo as empresas dos SI que as integram. Neste âmbito torna-se fundamental desenvolver software capaz de dar resposta às exigências do mercado e por esse facto é imperativo que o software actual seja devidamente modelado, para assim se diminuírem os desvios no cumprimento dos prazos de entrega do software e na ocorrência de erros de modelação e desenvolvimento que se traduzem em gastos acrescidos quando detectados no decorrer da fase de testes ou quando já estão em utilização (Kotonya, et al., 1996).

A UML dispõe de uma vasta lista de diagramas de modelação, cada um com características próprias, que se complementam e garantem a robustez de um sistema desenvolvido segundo uma correcta e experiente modelação. Por outro lado, com o mais recente contributo dos DA, uma vez distinguidos dos diagramas de Estados (Bhattacharjee, et

al., 2009), ficou notória a necessidade de se investir na sua compreensão e utilização na modelação de software.

A área da educação é vastíssima e esse facto leva-nos a pensar na necessidade de complementar os métodos de ensino actuais, com ferramentas de suporte e principalmente de treino prático. Aí entra o papel das TI, que tal como acontece nas organizações empresariais, também nas organizações educacionais, podem promover o seu desenvolvimento. Com base nestes factos propusemo-nos desenvolver um sistema de simulação dos DA, que contemplasse a visualização clara e objectiva das decisões e dos cenários alternativos que lhes sucedem, para permitir aos alunos efectuar testes sobre os diagramas que desenvolvem, não ficando só por conceitos teóricos e desenvolvimentos estáticos e sem a possibilidade de experimentação.

Tendo em conta o que foi escrito anteriormente relativamente ao ramo educacional e à classificação de software, Jucá (Jucá, 2006) classificaria esta aplicação como software educacional. Já Taylor (Taylor, 1980) com as suas definições clássicas mas perfeitamente adaptáveis a este contexto define três tipos de software educativo, "Tutor", "Ferramenta" e "Tutelado" e tendo em conta este sistema de classificação, a aplicação descrita neste estudo seria classificada de Ferramenta. Analisando um último sistema de classificação de software, Valente (Valente, 1998) definiria a aplicação desenvolvida como ferramenta de simulação.

Em relação à classificação dada por Valente, a nossa aplicação seria classificada como software de simulação, uma vez que um dos objectivos foi sempre tentar suprir as limitações apontadas por Navarro (Navarro, et al., 2005) em relação à maior parte do software, nomeadamente no que concerne à não interactividade e usabilidade.

Após a análise efectuada foi desenvolvido um software de simulação, e embora reconhecendo-lhe algumas limitações, pode ser usado como um complemento ao ensino de DA da UML.

5.2 Discussão dos resultados e principais contributos

O principal objectivo deste estudo consistiu em identificar de que forma uma aplicação poderia simular os DA de molde a esclarecer os alunos relativamente à sequência de execução, principalmente quando surgem nós de decisão e actividades concorrentes no processo de negócio.

Para isso, foi estruturado um conjunto de experiências com o objectivo de analisar os aspectos que à partida não fossem claros quanto à forma de implementação. Deste modo, respondendo às questões do investigador, 53% dos alunos, que previamente já tinham

visualizado um DA exemplo, responderam que a simulação apresentada lhes permitia uma maior percepção dos DA.

Relativamente à presença do som, as respostas não deixaram plano para dúvidas, isto é, respondendo à pergunta se a presença de som que descrevesse as actividades permitia uma melhor compreensão dos DA, 100% dos alunos responderam afirmativamente, isto porque o som funcionava como um reforço ao que surgia no ecrã.

No que diz respeito ao *fade-out* das actividades, isto é, ao desaparecimento das actividades à medida que estas iam ocorrendo, 25% dos alunos responderam que preferiam um *fade-out* parcial das actividades e os mesmos 25% responderam que preferiam um *fade-out* total sendo que 50% referiram que lhes era indiferente. No entanto, por uma questão de reforço, optámos por não incluir esta opção na aplicação, de maneira a que após realização de grandes diagramas construídos e simulados, os futuros alunos possam rever a simulação com todos os passos incluídos.

No que se refere à preferência da visualização dos cenários alternativos, isto é, se aquando do aparecimento de uma decisão, os cenários deveriam ser mostrados juntos ou separadamente, 29% dos alunos questionados responderam que preferiam cenários separados e 71% manifestaram a sua preferência por cenários juntos, dando como justificação o facto de poderem fazer a visualização de todos os cenários a qualquer momento e assim poderem esclarecer alguma dúvida que se mantivesse relativamente ao conjunto de cenários representados no DA.

O segundo objectivo, que está relacionado com o primeiro, diz respeito à construção da ferramenta propriamente dita, tendo em conta as conclusões tiradas até aqui.

Deste modo, como resultado final, pensamos que a aplicação criada poderá ser útil de forma a complementar a aprendizagem de DA da UML na medida em que o utilizador poderá:

- Perceber alguns conceitos dos DA que seriam mais complicados de perceber utilizando os métodos comuns de aprendizagem;
- Visualizar o processo de negócio descrito por um DA como se da sua execução se tratasse, na medida em que deixa de observar-se um diagrama estático e passa a percepcionar-se um conjunto de figuras que surgem no ecrã em momentos distintos de acordo com a sequência descrita;
- Corrigir potenciais erros presentes no DA e que não estavam previstos, mas que tenham sido identificados aquando da simulação do mesmo.No decorrer do desenvolvimento da ferramenta foram desenvolvidos dois protótipos cada um com características próprias.

O primeiro foi uma tentativa de simular o aparecimento das várias actividades associadas as seu actor, num espaço amplo do ecrã e que permitisse a conjugação das várias imagens de forma que o resultado final fosse uma simulação clara do processo representado no diagrama. Uma vez que este protótipo pretendia simular um DA, de acordo com o exemplo apresentado aos alunos aquando da realização das experiências, verificou-se à posteriori que este desenvolvimento seria complexo, já que as actividades iriam aparecer numa zona não pré determinada o que levaria, em casos de diagramas complexos e extensos, à acumulação de actividades no ecrã, dificultando assim o seu entendimento.

O segundo protótipo foi pensado para permitir visualizar qualquer tipo de diagrama, independentemente da sua extensão, já que a localização das actividades está pré-definida e as que lhes sucedem tomam uma posição inferior no ecrã fazendo-o deslocar-se no eixo dos “yy” para que a última actividade se mantivesse visível.

Os principais contributos que podemos reter da realização das experiências e dos protótipos da ferramenta são:

- As experiências já realizadas, de onde se podem retirar algumas conclusões que servem de base a um estudo futuro;
- Os problemas identificados no desenvolvimento do primeiro protótipo, que podem evitar atrasos num desenvolvimento subsequente e o trabalho desenvolvido no segundo protótipo, que poderá servir de ponto de partida de um projecto semelhante.

5.3 Desenvolvimento subsequente e propostas de trabalho futuro

Um projecto desta natureza considera-se sempre longo e em constante transformação, no entanto, tendo em conta os prazos de uma dissertação de mestrado não se afigurou possível a introdução de algumas funcionalidades que nos pareceram importantes.

Ao longo da realização deste estudo foram experienciadas algumas condicionantes e restrições à sua consecução. A noção da presença dessas restrições leva a considerar que as conclusões retiradas deste trabalho não podem deixar de ser consideradas como temporárias e susceptíveis de alterações através do desenvolvimento de um número maior e mais completos estudos.

Uma possível limitação tem a ver com a forma como as perguntas realizadas no decorrer das experiências foram respondidas pelos alunos, isto é, se houve tendência para responder às perguntas formuladas pelo investigador, de acordo com aquilo que o aluno associa a uma

maior aprovação. No entanto este ponto é sempre causador de algum erro, cabendo sempre ao investigador a formulação de questões de maneira a que o aluno sujeito às experiências não seja levado a uma resposta.

Outra limitação refere-se à não realização de experiências com alunos após a construção da aplicação de forma a validar a ferramenta construída. Assim, torna-se útil, como tarefa futura, incluir esta verificação de modo a que se consiga percepcionar o seu real valor.

Tendo em conta os resultados finais das primeiras experiências realizadas, pensamos que a introdução de som à medida que as actividades vão surgindo se torna um elemento de melhor compreensão para quem aprende este tipo de diagramas.

Também a introdução da funcionalidade de opção de visualização de todos os cenários após uma decisão, nos parece ser uma decisão correcta, isto é, quando surge um ponto de decisão, permitir ao utilizador optar por visualizar todos os cenários guardados pelas instruções booleanas, evitando a necessidade de retomar o processo para seleccionar outra condição de guarda e prosseguir com outro cenário. No entanto, a possível implementação desta opção terá que ter em conta a forma como os diferentes cenários serão mostrados, já que na ferramenta desenvolvida, verificou-se alguma complexidade em mostrar diferentes cenários na mesma simulação, dificultando também a compreensão do processo de negócio apresentado. Assim, na aplicação desenvolvida, foi imperativo mostrar apenas o cenário seleccionado pelo utilizador, devido à percentagem de preenchimento do ecrã e à sequência de amostragem das actividades.

5.4 Conclusão

Chegando ao fim deste trabalho de investigação/desenvolvimento é chegado o momento de referir algumas conclusões sobre o estudo.

O esforço desenvolvido no desenvolvimento deste projecto torna-se ainda mais compensatório sabendo que em todo o mundo, (Jones, 2007):

- São realizados mais de dois por cento de alterações mensais aos requisitos após a fase de especificação dos mesmos.
- São encontrados mais de seis defeitos por ponto de função, em média.
- A eficiência é inferior a 35 % na remoção dos defeitos antes do início dos testes.
- A eficiência é inferior a 85 % na remoção dos defeitos antes da disponibilização do software.

Em suma, não consideramos que o trabalho realizado possa alterar em grande escala estes números, no entanto, saber que a aplicação desenvolvida ajudaria a uma melhor compreensão dos DA por parte dos alunos, já torna todo o trabalho realizado numa enorme compensação.

Consideramos que este estudo está inacabado, no sentido de lhe perspectivarmos já uma continuidade para outros projectos. Desta forma, com humildade, pensamos que este poderá ser um ponto de partida para o desenvolvimento de novas aplicações, utilizando a simulação, para os vários diagramas da UML.

Bibliografia

Alava S. Ciberespaço e formações abertas: Rumo a novas práticas educacionais? [Livro]. - Porto Alegre : ArtMed, 2002.

Almeida J. C. F. Em defesa da investigação-acção. [Artigo] // Sociologia. - Oeiras : Publicações Europa-América, 2001. - Vol. 37.

Arlow J. e Neustadt I. UML and the Unified Process - Practical Object-Oriented Analysis & Design [Livro]. - Boston : Addison-Wesley, 2002.

Askari M. R. e Davis R. Development and Application of Computer Software for Simulation of Vibration Analysis in Education [Jornal] // Simulation '98. - 1998. - pp. 329-337.

Axelrod R. Advancing the Art of Simulation in the Social Sciences [Artigo] // Journal of the Japan Society for Management Information. - Tóquio, Japão : Science Links Japan, 1997. - 2. - Vol. 3.

Banks J. Software for Simulation. [Conferência] // Proceedings of the 1994 Winter Simulation Conference. - Lake Buena Vista, FL : IEEE, 1994.

Barros J. P. e Gomes L. Towards the Support for Crosscutting Concerns in Activity Diagrams : a Graphical Approach [Conferência] // 4th AOM Workshop at UML. - San Francisco, CA : [s.n.], 2003.

Baskerville R. Investigating information systems with action research. [Jornal]. - Omaha, NE : Communications of the Association for Information Systems, 1999. - Vol. 2.

Bell D. UML basics Part II: The activity diagram [Relatório]. - [s.l.] : Rational Edge, 2003.

Bhattacharjee A. K. e Shyamasundar R. K. Activity Diagrams : A Formal Framework to Model Business Processes and Code Generation [Jornal]. - Zurich : Journal of Object Technology, 2009. - Vol. 8.

Booch G., Jacobson I. e Rumbaugh J. The Unified Modeling Language for Object-Oriented Development v. 0.9 [Relatório]. - [s.l.] : Rational Software Corp., 1996.

Booch G., Rumbaugh J. e Jacobson I. The Unified Modeling Language User Guide: Second Edition [Livro]. - Boston : Addison-Wesley, 2005.

Boticario J. G. e Santos O. C. Issues in Developing Adaptive Learning Management Systems for Higher Education Institutions [Conferência] // Workshop on Adaptive Learning and Learning Design. - Dublin, Irlanda : [s.n.], 2006.

Brugge B. e Dutoit A. Object-Oriented Software Engineering: Conquering Complex and Changing Systems. [Livro]. - Upper Saddle River, NJ : Prentice-Hal, 2001.

Bunge M. ”Teoria Estática”. La Investgación Científica: su Estrategia y su Filosofia. [Livro]. - Barcelona : Ariel, 1980. - p. 413.

Callahan D. e Pedigo B. Educating Experienced IT Professionals by Addressing Industry's Needs. [Artigo] // IEEE Software. - Birmingham, AL : IEEE, 2002. - 5. - Vol. 19.

Carvalho P. C. M. e Jucá S. C. S. Programa didático de dimensionamento de sistemas fotovoltaicos autónomos [Conferência] // Congresso Brasileiro de Ensino de Engenharia. - Rio de Janeiro, Brasil : COBENGE, 2003.

Chang E., Gautama E. e Dillon T.S. Extended Activity Diagrams for Adaptive Workflow Modelling. [Conferência] // ISORC 2001. - Magdeburg, Alemanha : IEEE, 2001.

Chiavenato I. Iniciação ao Planejamento e Controle de Produção [Livro]. - São Paulo : McGrawHill, 1990.

Claypool K. e Claypool M. Teaching Software Engineering Through Game Design [Conferência] // Proceedings of the Tenth Annual Conference on Invention and Technology in Computer Science Education. - New York, NY : ACM, 2005.

Cohen L., Manion L. e Morrison K. Research Methods in Education [Livro]. - Londres : RoutledgeFalmer, 2000.

Collofello J. S. [et al.] Using Software Process Simulation to Assist Software Development Organizations in Making Good Enough Quality Decisions. [Conferência] // Summer Computer Simulation Conference. - Edimburgo, Escócia : SCSC98, 1998.

Conrow P. S. e Shishido E.H. Implementing risk management on software intensive projects [Jornal] // IEEE Software 14. - 1997. - pp. 83–89.

Crandal R. l. e Jackson C. The $500 Billion Opportunity: The Potential Economic Benefit of Widespread Diffusion of Broadband [Jornal]. - Washington, D.C. : Criterion Economics, 2001.

Dick B. Postgraduate programs using action research [Artigo] // The Learning Organization. - Bingley, UK : Emerald, 2002. - 4. - Vol. 9.

Dobing B. e Parsons J. How UML is used [Artigo] // Commun. - [s.l.] : ACM, 2006. - 5. - Vol. 49. - pp. 109-113.

Doukidis G. I. e Paul R. J. A Survey of the Application of Artificial Intelligence Techniques within the OR Society. [Artigo] // The Journal of the Operational Research Society. - Birmingham, UK : [s.n.], 1990. - 5. - Vol. 41.

Drappa A. e Ludewig J. Simulation in Software Engineering Training [Jornal] // Proceedings of the 22nd International Conference on Software Engineering: ACM. - 2000. - pp. 199-208.

Educação Ministério da Plano de acção anual para as tecnologias da informação e comunicação [Relatório]. - [s.l.] : Ministério da Educação, 2007.

Eichelberger H. e Gudenberg J. UML Class Diagrams – State of the Art in Layout Techniques [Conferência] // ICSM'2003. - Amesterdão : [s.n.], 2003.

Eischen K. Software Development: An Outsider's View [Artigo] // Computer. - Santa Cruz, CA : IEEE Computer Society, 2002. - 5. - Vol. 35.

Eshuis H. Semantics and Verification of UML Activity Diagrams for Workflow Modelling [Relatório] : Tese PhD / Department of Computer Science ; University of Twente. - Twente : University of Twente, 2002.

Eshuis R. e Wieringa R. A Real-Time Execution Semantics for UML Activity Diagrams [Conferência] // FASE 2001. - Génova, Itália : Springer, 2001.

Furneaux C. How does Information Technology impact the methods, potential and purpose of education. [Conferência] // ETL Conference. - Brisbane, Australia : ETL, 2004.

Galli E. A. Conocimiento Tecnológico, Educacion y Tecnologia. [Artigo] // Revista Iberoamericana de Innovaciones Educativas. - Buenos Aires : [s.n.], 1993. - 12. - Vol. 5.

Gamito R., Cunha L.A. e Abreu S. Modelação de Workflows com UML e Ferramentas Declarativas [Conferência] // XATA2007 — XML: Aplicações e Tecnologias Associadas. - Lisboa : [s.n.], 2007.

Gnatz M. [et al.] A Practical Approach of Teaching Software Engineering. [Conferência] // Proceedings of the 16th Conference on Software Engineering Education and Training. - Madrid, Espanha : IEEE Computer Society, 2003.

Gogg T. J. e Mott J. R. A. Introduction to simulation. [Conferência] // Conference Proceedings of the 25th conference on Winter simulation. - Los Angeles, CA : Winter Simulation, 1993.

Goguen J. e Jirotka M. Requirements Engineering: Social and Technical lssues. [Livro]. - London : Academic Press, 1994.

Gonçalves R. Ciência, Pós-Ciência, Metaciência [Livro]. - Lisboa : Terramar, 1997. - 2ª edição.

Gouveia L. B. Negócio Electrónico – Conceitos e Perspectivas de Desenvolvimento [Livro]. - Porto : SPI – Sociedade Portuguesa de Inovação, 2006.

Gramigna M. R. M. Jogos de Empresa. [Livro]. - São Paulo : Makron Books, 1993.

Hammer D. K., Hanish A. A. e Dillon T. S. Modeling Behavior and Dependability of Object-Oriented Real-TimeSystems [Artigo] // Journal of Computer Systems Science & Engineering. - [s.l.] : WASET, 1998. - 3. - Vol. 13.

Hlupic V. Business Process Re-engineering and Simulation: Bridging the Gap [Conferência] // Proceedings of the European Simulation Multiconference ESM'98. - Manchester,UK : SCS, 1998.

Hlupic V. Simulation Software: an Operational Research Society Survey of Academic and Industrial Users [Conferência] // Proceedings of the 2000 Winter Simulation Conference. - Orlando, FL : IEEE, 2000. - pp. 1676-1683.

Hugon M. A. e Seibel C. Recherches impliquées, recherches action: le cas de l'éducation. [Conferência] // Colloque organisé par l'Institut National de Recherche Pédagogique. - Bruxelas : De Boeck-Wesmael, 1988.

Hulme M. e Locasto M. Using the Web to enhance and transform education [Artigo] // ACM Crossroads. - [s.l.] : ACM, 2003. - 1. - Vol. 10.

IEEE IEEE Computer Dictionary - Compilation of IEEE Standard Computer Glossaries, 610-1990 [Relatório]. - [s.l.] : IEEE, 1991.

Jackson M. e Zave P. Deriving Specifications from Requirements: An Example. [Conferência] // Proceedings of the 17th ICSE. - Seattle : IEEE, 1995.

Jones C. Conflict and Litigation Between Software Clients and Developers [Relatório]. - Boston, MA : Software Productivity Research, Inc., 2007.

Jucá S. C. S. A relevância dos softwares educativos na educação profissional. [Artigo] // Ciências & Cognição. - Ceará, Brasil : ICC, 2006. - 3. - Vol. 8.

Knezek G. A., Rachlin S. L. e Scannell P. A Taxonomy for Educational Computing,, 28, 4 (1988), 15-19. [Artigo] // Educational Technology. - [s.l.] : IEEE, 1988. - 4. - Vol. 28.

Kotonya G. e Sommerville I. Requirements Engineering with Viewpoints [Jornal] // BCS/IEE Software Engineering Journal 11. - 1996. - pp. 5-18.

Kuhne G. W. e Quigley B. A. Understanding and Using Action Research in Practice Settings. [Artigo] // New Directions for Adult and Continuing Education. - San Francisco, CA : Jossey-Bass, 1997. - 73.

Lethbridge T. C. What Knowledge is Important to a Software Professional?. [Artigo] // IEEE Computer. - Ottawa, Canadá : IEEE, 2000. - 5. - Vol. 33.

Lieberman B. Using UML Activity Diagrams for the Process View. [Relatório]. - [s.l.] : Rational Edge, 2001.

McKeown P. G. e Leitch R. A. Management Information Systems: Managing with Computers. [Jornal]. - New York : The Dryden Press, 1993.

Metzger P. W. e Boddie J. Managing a Programming Project [Livro]. - Englewood Cliffs, NJ : Prentice Hall, 1996. - 3.

Meyer B. Software Engineering in the Academy [Artigo] // Computer. - Los Alamitos, CA : IEEE, 2001.

Mingins C. [et al.] How We Teach Software Engineering [Artigo] // The Journal of Object - Oriented Programming. - Melbourne, Australia : SIGS Publications, 1999. - 9. - Vol. 11.

Monadjemi A. e Ahmadi A. How will Computer Aided Learning Develop [Conferência] // 8th Iranian Students Seminar in Europe. - Manchester, UK : UMIST, 2001.

Mota A. [et al.] FEUP [Online]. - 2004. - 15 de 5 de 2009. - http://paginas.fe.up.pt/~ei02084/artigo.pdf.

Myers M. Qualitative Research in Information Systems, 21, Junho 1997 [Artigo] // MIS Quarterly. - Minneapolis, MN : Society for Information Management and The Management Information Systems Research Center, 1997. - 2. - Vol. 21.

Naur P. e Randell B. Software Engineering: Report of a conference sponsored by the NATO Science Committee [Conferência] // Scientific Affairs Division, NATO. - Garmisch : [s.n.], 1968.

Navarro E. O. e Hoek A. v. d. Design and Evaluation of an Educational Software Process Simulation Environment and Associated Model [Conferência] // Proceedings of the 18th Conference on Software Engineering Education and Training. - Ottawa, Canadá : IEEE Computer Society, 2005.

Navarro E. O., Baker A. e van der Hoek A. An Experimental Card Game for Teaching Software Engineering. [Conferência] // Proceedings 16th Conference on Software Engineering Education and Training. - Madrid, Espanha : IEEE, 2003.

Nikoukaran J., Hlupic V. e Paul R. J. Criteria for Simulation Software Evaluation [Jornal] // Proceedings of the 1998 Winter Simulation Conference. - 1998. - pp. 399-406.

Nuseibeh B. e Easterbrook S. Requirements Engineering: A Roadmap [Conferência] // Future of Sofware Engineering. - Limerick, Irlanda : ACM, 2000.

Oblinger D. G. e Rush, S. C., The Learning Revolution The Challenge of Information Technology in the Academies [Livro]. - Bolton : Anker, 1997.

Oliveira C. C., Menezes E. I. e Moreira M. Ambientes Informativos de Aprendizagem: produção e avaliação de software educativo. [Livro]. - Campinas : Papirus, 2001.

OMG OMG UML 1.3 [Online] // Object Management Group. - 8 de 6 de 1999. - 5 de 4 de 2009. - http://www.omg.org.

OMG UML Revision Taskforce, “OMG UML Specification v.1.5” [Online]. - Object Management Group, 1 de 3 de 2003. - 12 de 4 de 2009. - http://www.omg.org.

Orlikowski W. J. e Robey D. Information technology and the structuring of organizations [Artigo] // Information Systems Research. - Cambridge, MA : HighWire Press, 1991. - 2. - Vol. 2.

Papert S. A Família em Rede. [Livro]. - Lisboa : Relógio d' Água, 1997.

Pérez Serrano G. Investigacion Cualitativa I: Retos e Interrogantes: Metodos [Livro]. - Madrid : La Muralla, 1994.

Pinto C. N. Actas da III Conferência Internacional de Tecnologias da Informação e Comunicação na Educação [Conferência] // Conferência Internacional de Tecnologias da Informação e Comunicação na Educação. - Braga : Universidade do Minho, 2003.

Rumbaugh J., Booch G. e Jacobson I. The Unified Modeling Language User's Guide [Livro]. - Boston : Addison-Wesley, 1999.

Shaw M. Software Engineering Education: A Roadmap. [Artigo] // The Future of Software Engineering. - Nova York, NY : ACM Press, 2000.

Siebenhaller M. e Kaufmann M. Drawing Activity Diagrams [Conferência] // SOFTVIS 2006. - Brighton, UK : Association for Computing Machinery, 2006.

Silva C. R. Epistemologia do Conhecimento Tecnológico como base de Geração, aplicação e difusão de Tecnologia. [Jornal]. - Fortaleza : IDEIAS Fortaleza, 1996.

Skrien D. CPU Sim 3.1: A tool for simulating computer architectures for computer organization classes. [Jornal] // ACM Journal of Educational Resources in Computing. - 2001. - pp. 46-59.

Sommerville I. Software Engineering [Livro]. - Harlow : Addison Wesley, 2006. - 8.

Souto P. M. Educar en Ciudades de la Ciencia [Artigo] // A página da educação. - Porto : Profedições, 2008. - XVII.

Sparx Using UML Part Two – Behavioral Modeling Diagrams [Relatório]. - [s.l.] : Sparx Systems, 2007.

Suppes P. The uses of computers in education [Jornal]. - São Francisco, CA : Scientific American, 1966.

Surendran K. e Young F. H. Teaching Software Engineering in a Practical Way [Conferência] // National Advisory Committee on Computing Qualifications. - Wellington, Nova Zelândia : NACCQ, 2000.

Taylor R. P. The Computer in the School: Tutor, Tool, Tutee. [Livro]. - New York : Teachers College Press, 1980.

Valente J. A. Diferentes usos do Computador na Educação [Artigo] // Computadores e Conhecimento: Repensando a Educação. - Campinas : Nied, 1998.

van Vliet H. Software Engineering: Principles and Pratices [Livro]. - Glasgow : Wiley, 2007. - 3.

Varajão J. E. Q. Arquitectura da Gestão de Sistemas de Informação [Livro]. - Lisboa : FCA, 2005.

Wilson J. R. A survey of research on the simulation startup problem. [Jornal] // SIMULATION. - 1978. - pp. 55-58.

Woolgar S. Laboratory studies: a comment on the state of the art. [Artigo] // Social studies of science. - Londres : Sage, 1982.

Anexos

Anexo I - Pedido de orçamento (construção civil)

Enunciado de problema

- Quando um cliente pretende um orçamento de obras de construção civil dirige-se ao balcão de atendimento da sede da empresa UTADCONSTROI para efectuar um pedido de proposta de orçamento, o qual entrega ao funcionário comercial.

- Se o pedido de proposta de orçamento for igual ou inferior a 500€, o próprio funcionário comercial prepara a proposta de orçamento, mas se o pedido for superior a 500€, este é encaminhado para o director comercial para este preparar a proposta de orçamento.

- Depois da proposta de orçamento estar pronta, esta é sempre enviada ao cliente pelo funcionário comercial, independentemente de quem a preparou.

Anexo II – Pedido de justificação de faltas (universidade)

Enunciado de problema

- Quando um aluno pretende justificar as suas faltas dirige-se à secretaria da universidade.

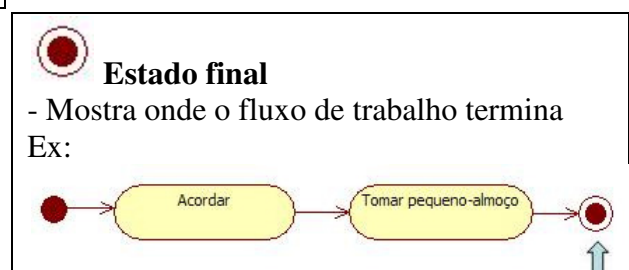
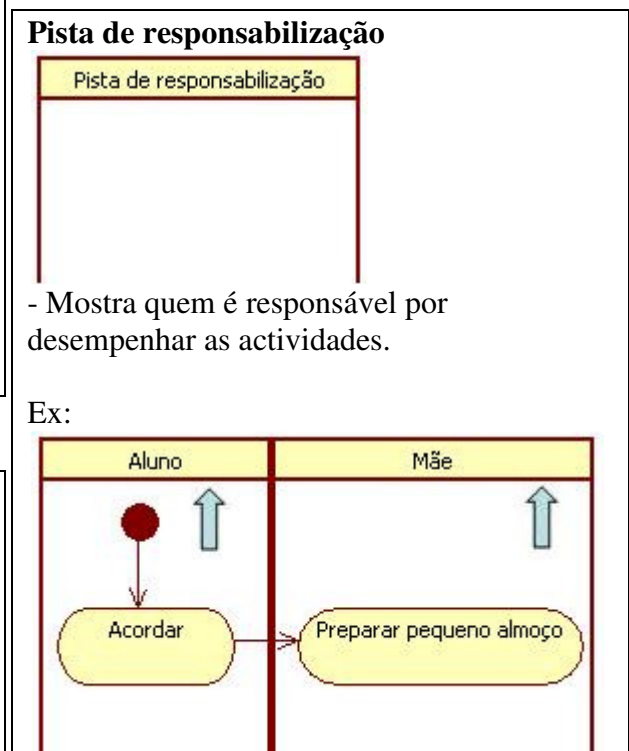
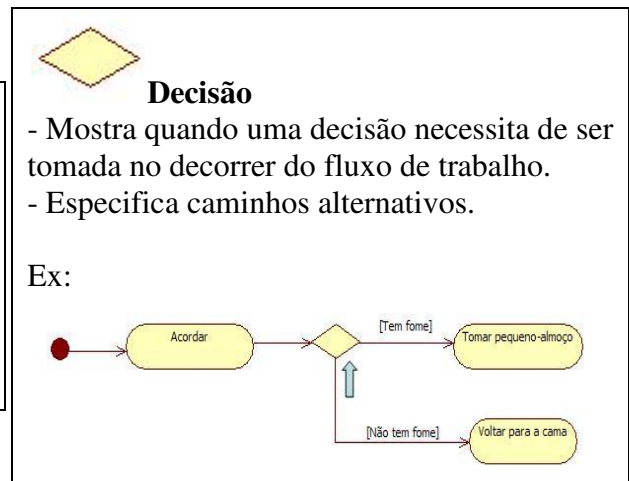
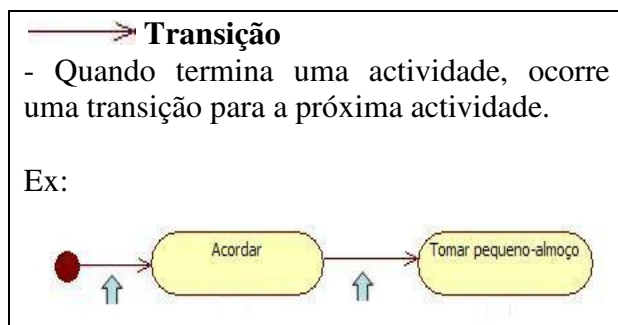
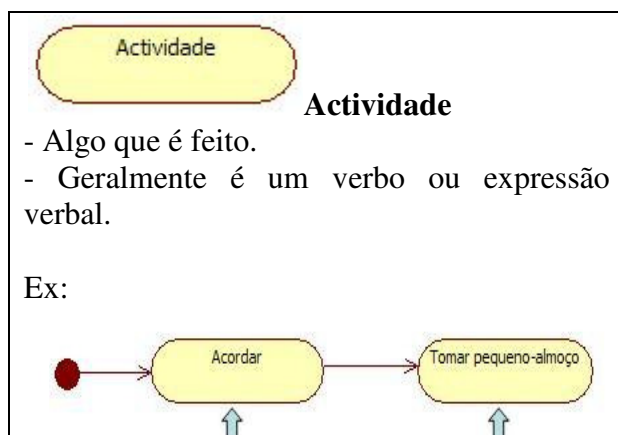
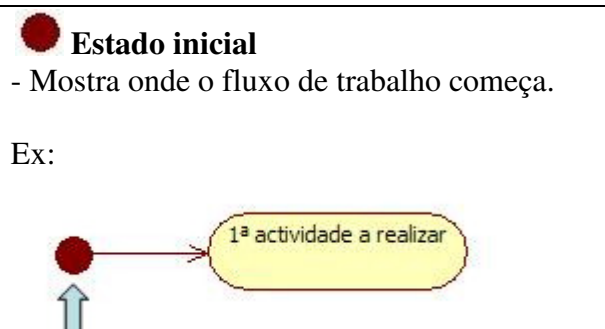
- A justificação é entregue ao funcionário da secretaria, o qual analisa a justificação.

- Se a justificação não suscitar dúvidas, é o próprio funcionário da secretaria que decide a aceitação da mesma, mas se a justificação suscitar dúvidas esta é encaminhada para o professor da unidade curricular para que seja este a decidir a aceitação ou não da justificação. No caso da decisão ser efectuada pelo professor, este informa o funcionário da aceitação ou não da justificação.

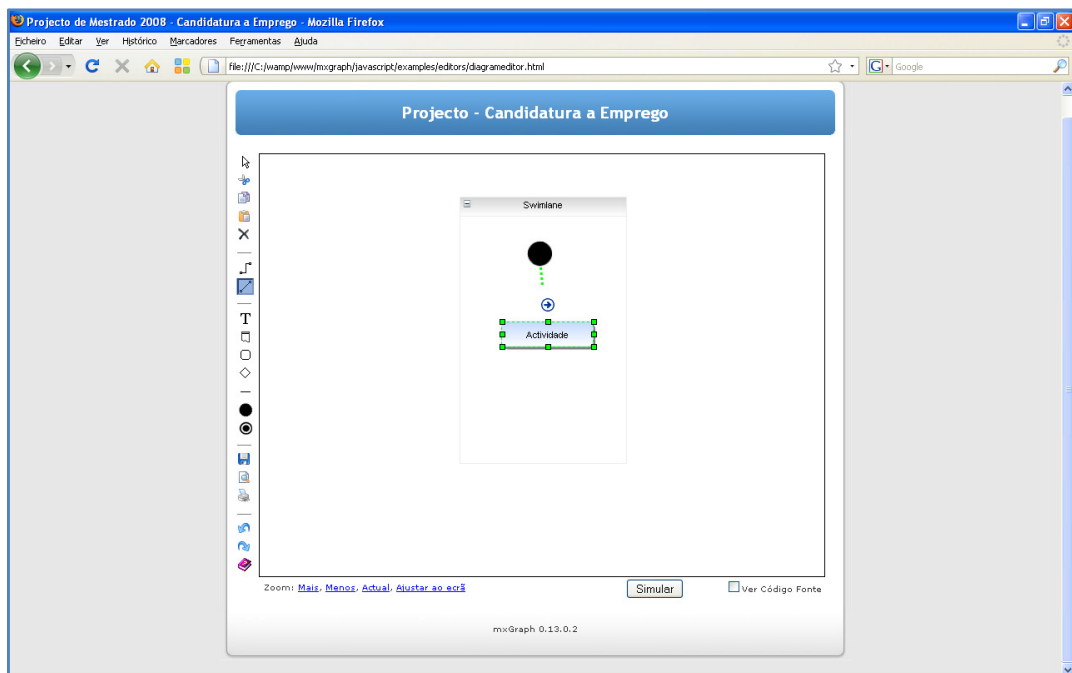
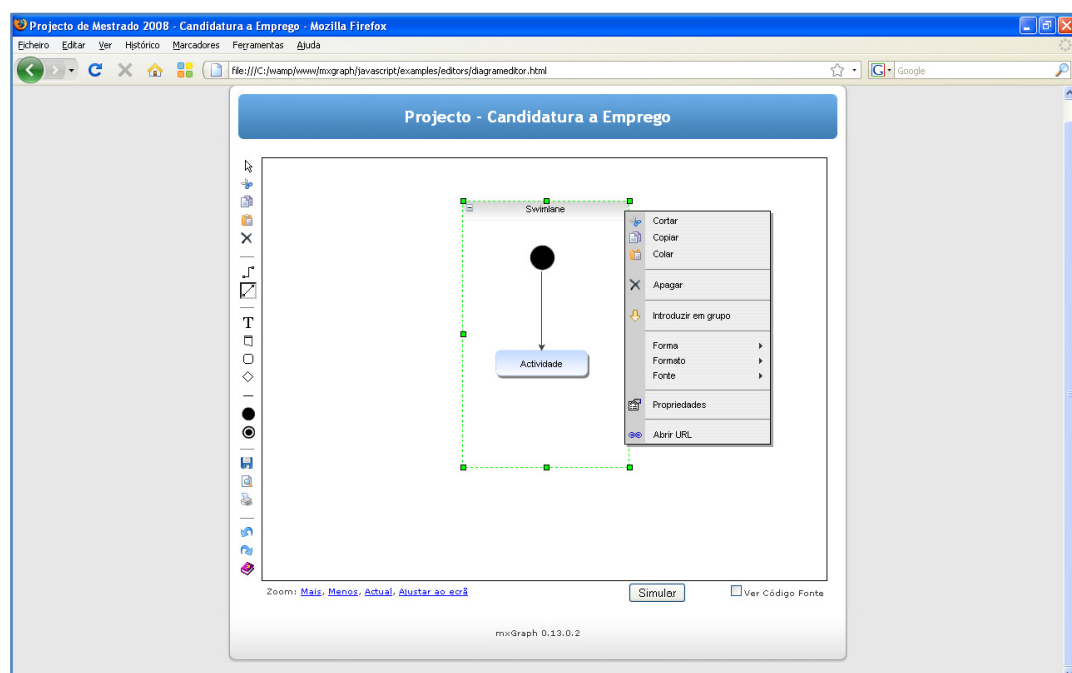
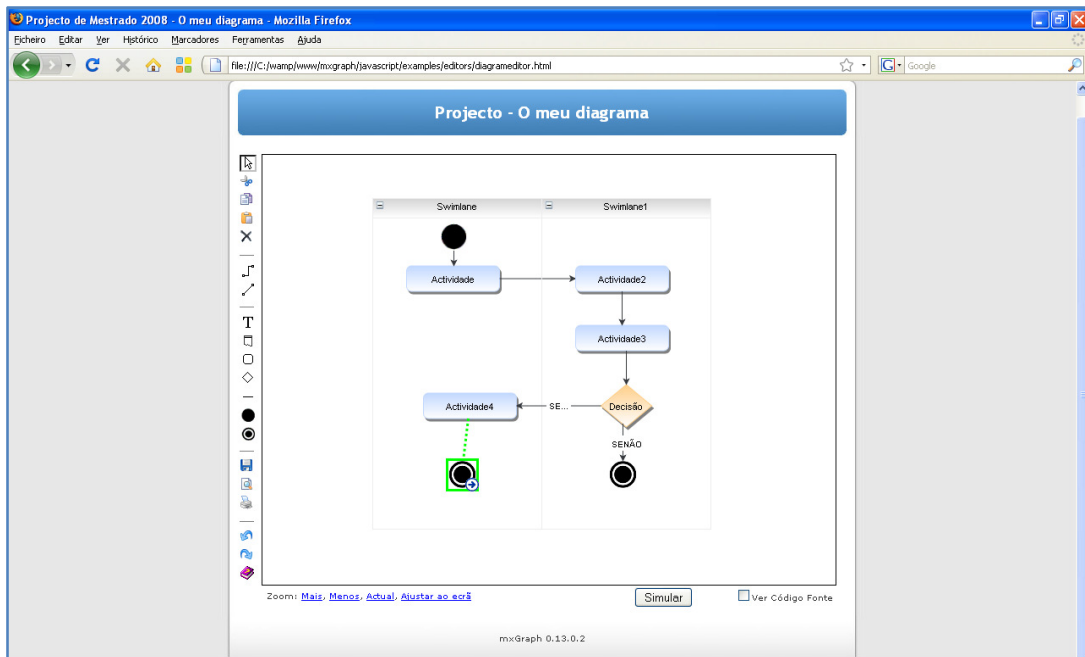
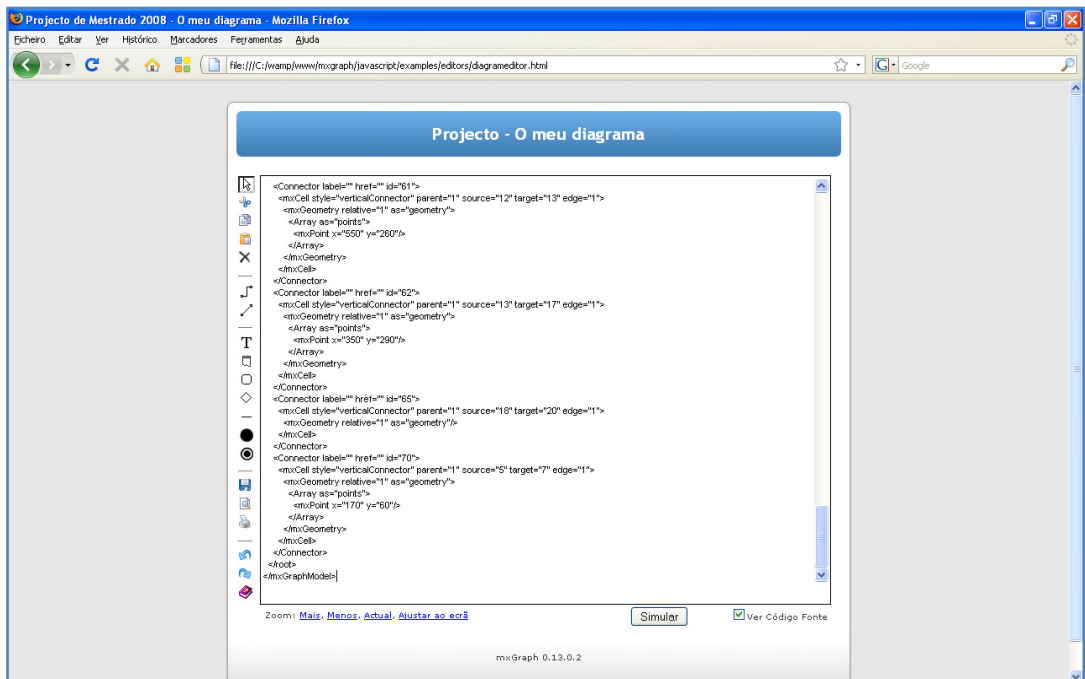
- Depois de tomada a decisão, cabe ao funcionário de secretaria fazer o seu registo formal.

Anexo III – Manual sintético

Os diagramas de actividades possibilitam representar actividades e a sequência de realização dessas mesmas actividades (fluxo de trabalho).

Componentes:

Anexo IV – Editor de diagramas de actividades da UML

Neste anexo apresentam-se várias imagens que representam momentos diferentes da criação de um diagrama de actividades.

1. Início da construção de um diagrama de actividades.

2. Alteração de propriedades do diagrama de actividades.

3. Construção de um diagrama com decisões e cenários alternativos.

4. Visualização do código XML do diagrama de actividades criado.

Anexo V – Ferramenta de simulação de diagramas de actividades da UML

Neste anexo apresentam-se várias imagens que representam momentos diferentes da ferramenta de simulação desenvolvida.

1. Início da simulação do diagrama de actividades criado.

2. Final da simulação do diagrama de actividades criado.